ADOBE PHOTOSHOP CS6

JAVASCRIPT SCRIPTING REFERENCE

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Contents

1 Introduction

This reference describes the objects and methods in the Adobe® Photoshop® CS6® JavaScript™ type library. A companion document, *Photoshop CS6 Scripting Guide*, describes basic scripting concepts and the Photoshop object model. This document provides reference details of the Photoshop object model, and additional information on JavaScript-specific features.

Adobe Photoshop CS6 uses ExtendScript, Adobe's extended implementation of JavaScript. See JavaScript support in Adobe Photoshop CS6 for additional information.

This book contains the following sections:

- This introduction, which describes scripting support in Adobe Photoshop CS6, and lists changes to the JavaScript interface since the previous release.
- JavaScript Object Reference, which provides a complete reference for all Photoshop DOM objects and commands.
- Scripting Constants, which lists all enumerations used in the Photoshop type library.

JavaScript support in Adobe Photoshop CS6

For a JavaScript file to be recognized by Photoshop as a valid script file, it must use either a `.js` or a `.jsx` extension.

On the Mac OS, there is no difference in the way scripts with the two extensions function. On Windows, if the script files is opened from inside Photoshop, there is no difference between using the `.js` and `.jsx` extension. However, if the script is launched by double-clicking on it, a script with the `.js` extension is interpreted with the Microsoft JScript engine, and it cannot launch Adobe Photoshop CS6. For Windows, using the `.jsx` extension is preferrable, since it interprets the script with the ExtendScript engine.

All of the Adobe Creative Suite 6 applications, including Adobe Photoshop CS6, use ExtendScript, Adobe's extended implementation of JavaScript. ExtendScript files are distinguished by the `.jsx` extension. ExtendScript offers all standard JavaScript features, plus additional features and utilities, such as:

- A debugging environment (the ExtendScript Toolkit)
- A localization utility
- Tools that allow you to combine scripts and direct them to particular applications
- Platform-independent file and folder representation

Many of the JavaScript objects and methods use objects defined in ExtendScript, such as the `File` object, the `Folder` object, and the `UnitValue` object. For that reason, using the `.jsx` extension for your script files is preferable.

For details of these and additional features, see the *JavaScript Tools Guide CS6*. This document is installed with Creative Suite 6 applications at these locations:

- In Windows:

  `C:\Program Files\Adobe\Adobe Utilities\ExtendScript Toolkit CS6\SDK`

- In Mac OS:

  `Applications/Utilities/Adobe Utilities/ExtendScript Toolkit CS6/SDK`

The latest versions of this document and of the ExtendScript Tookit, can also be downloaded from Adobe Developer Center, http://www.adobe.com/devnet/.

Executing scripts

The Adobe Photoshop CS6 interface includes a Scripts menu (**File > Scripts**) which provides quick and easy access to your JavaScripts. Scripts can be listed directly as menu items that run when you select them, or you can navigate to and run any JavaScript in your file system.

If Adobe Photoshop CS6 encounters an error during script execution, it displays the error message.

Installing scripts

To install a JavaScript in the Scripts menu, place it in the Scripts folder (**Photoshop CS6/Presets/Scripts**). The names of the scripts in the Scripts folder, without the file name extension, will be displayed in the Scripts menu. Any number of scripts may be installed in the Scripts menu.

Scripts added to the Scripts folder while Adobe Photoshop CS6 is running will not appear in the Scripts menu until the next time you launch the application.

All scripts found in the Scripts folder and sub-folders are displayed at the top level of the **File > Scripts** menu. The addition of sub-folders does not add a hierarchical organization to the Scripts menu.

Executing other scripts

The **Browse** item at the end of the **Scripts** menu (**File > Scripts > Browse**) allows you to execute scripts which are not installed in the Scripts folder. You can also use Browse to select scripts installed in the Scripts folder after the application was last launched.

Selecting **Browse** displays a file browser dialog which allows you to select a script file for execution. Only `.js` or `.jsx` files are displayed in the browse dialog. When you select a script file, it is executed the same way as an installed script.

Startup scripts

On startup, Adobe Photoshop CS6 executes all `.jsx` files that it finds in the startup folders.

- On Windows, the startup folder for user-defined scripts is:

  `C:\Program Files\Common Files\Adobe\Startup Scripts CS6\Adobe Photoshop`

- On Mac OS, the startup folder for user-defined scripts is:

  `~/Library/Application Support/Adobe/Startup Scripts CS6/Adobe Photoshop`

If a script is meant to be executed only by Adobe Photoshop CS6, it must include code such as the following:

```
if( BridgeTalk.appName == "photoshop" ) {
    //continue executing script
}
```

For additional details, see the *JavaScript Tools Guide CS6*.

Object model usage and naming

The JavaScript API follows JavaScript naming conventions in that all classes (object types) begin with uppercase letters and have mixed case. Typically, in JavaScript, you instantiate classes using the `new` operator:

```
new ClassName();
```

However, in the Photoshop Object Model, it is often not necessary to do this. Major object types are collected into collection classes; for example, a list of `Document` objects is contained in a `Documents` collection object. You then access the collection object through a corresponding collection property in its container in the object hierarchy.

For example, the collection of all open documents is contained in the top-level `Application` object. You can access this through the global variable `app`, or simply reference its properties directly at the top level:

```
app.documents[0] // get the first loaded documented
documents[0] // this is the same
```

A collection property has the same name as the collection object, but begins with lowercase. For example, a `Document` contains a collection of `LayerSets`, and a `LayerSet` contains a collection of `ArtLayers`. To access one `ArtLayer` object in a set:

```
var myLayer = activeDocument.layerSets[0].artLayers[0];
```

The collections, as in this example, can be treated as arrays, which is useful for iteration. They also provide methods to create their contained objects, and to access them by name:

```
var newLayer = activeDocument.artLayers.add(); // Create a new ArtLayer object
newLayer.name = "My Layer"; // name it for later reference
...
var layerRef = activeDocument.artLayers.getByName("My Layer");
```

Some objects, such as the `Font` objects contained in the `app.fonts` collection, are created by the application, and never by your scripts.

Your scripts do use the JavaScript `new` operator to create helper objects, such as those that encapsulate a set of options for opening or saving a document in a particular format:

```
var opts = new PDFOpenOptions();
opts.page = 10;
app.open(myPDFFile, opts);
```

Changes since earlier versions—CS6 changes

The following changes have been made to the JavaScript object model and language support in Adobe Photoshop CS6:

- New methods and properties:
  - `getLargeInteger` has been added to the `ActionDescriptor` and `ActionList` objects.
  - `putLargeInteger` has been added to the `ActionDescriptor` and `ActionList` objects.
  - `compression` method has been added to the `PNGSaveOptions` object.
- Modified enumerations:
  - `LARGEINTEGERTYPE` has been added for the `DescValueType` enumeration.
  - `EXTRALARGE` and `HUGE` have been added for the `FontPreviewType` enumeration.

2 JavaScript Object Reference

The Photoshop objects (the JavaScript type library for Adobe® Photoshop® CS6) are presented alphabetically and in tabular format in this chapter. Sample code for several object model classes is given to help illustrate the syntax as well as usage of the object class.

Object descriptions

Object properties and methods are described in separate tables for each object. The following sections describe the conventions used in these descriptions.

Properties notation

The Properties table for an object lists the following:

- The properties defined in each object.
- The value type for each property.

  When the value type is a constant or another object, the value is a hypertext link to the listing for that constant or object.
- The property's input status: read-only or read-write.
- A description that explains what the property does.

Property	Value type	What it is
displayDialogs	DialogModes	Read-write. The dialog mode for the application, which controls what types of dialogs should be displayed when running scripts.

For constants, like `DialogModes` in the sample, click the link to go to the table that shows allowed values. Constants are represented by objects, and allowed values are properties of those objects. Specify a constant value in the form *ConstantName.VALUE*. For example:

```
app.displayDialogs = DialogModes.ERROR;
```

Methods notation

The Methods table for an object lists the following:

- The method name.
- The parameters list.
- The parameter value types, on lines corresponding to each parameter.
- Return value type
- A description of what the method does, and further descriptions of parameters, if needed.

Method	Parameter type	Returns	What it does
crop (bounds [, angle] [, width] [, height])	array of 4 UnitValue number UnitValue UnitValue		Crops the document. The `bounds` parameter is an array of four coordinates for the region remaining after cropping, [left, top, right, bottom].

When a parameter type or return value is a constant or another object, the value is a hypertext link to the listing for that constant or object.

Parameters can be required or optional. Optional parameters are indicated in the table by square brackets ([]). In the example, the first parameters, *bounds*, is required. The remaining parameters are all optional.

You must pass a value for each required parameter. You can leave out optional parameters if there are no remaining values to pass; however, if you wish to use the default value for any optional parameter that is not the last one specified, pass `undefined` as a placeholder. You must enter the values in the order they are listed, so that the JavaScript compiler knows which value you are entering.

For example, the following passes only the required parameter (using a previously-defined variable for the bounding region):

```
app.activeDocument.crop( myRegion );
```

The following skips the *angle* parameter, specifies the *width* value, and omits the final *height* value:

```
var myWidth = new UnitValue( "500 pixels" );
app.executeAction( myRegion,undefined,myWidth );
```

ActionDescriptor

This object provides a dictionary-style mechanism for storing data as key-value pairs. It can be used for low-level access into Photoshop. See an example of this usage in 'Selection sample script' on page 168.

Many configuration files use serialized action descriptors to represent their data. It is used, for example, to encapsulate playback options in Application.playbackParameters, and is returned by Application.getCustomOptions().

Properties

Property	Value type	What it is
count	number	Read-only. The number of keys contained in the descriptor.
typename	string	Read-only. The class name of the referenced `actionDescriptor` object.

Methods

Method	Parameter type	Returns	What it does
clear ()			Clears the descriptor.
erase (key)	number		Erases a key from the descriptor.
fromStream (value)	string		Creates a descriptor from a stream of bytes; for reading from disk.
getBoolean (key)	number	boolean	Gets the value of a key of type boolean.
getClass (key)	number	number	Gets the value of a key of type class.
getData (key)	number	string	Gets raw byte data as a string value.
getDouble (key)	number	number	Gets the value of a key of type double.
getEnumerationType (key)	number	number	Gets the enumeration type of a key.
getEnumerationValue (key)	number	number	Gets the enumeration value of a key.
getInteger (key)	number	number	Gets the value of a key of type integer.
getKey (index)	number	number	Gets the ID of the *N*th key, provided by index.

Method	Parameter type	Returns	What it does (Continued)
getLargeInteger (key)	number	number	Gets the value of a key of type large integer.
getList (key)	number	ActionList	Gets the value of a key of type list.
getObjectType (key)	number	number	Gets the class ID of an object in a key of type object.
getObjectValue (key)	number	ActionDescriptor	Gets the value of a key of type object.
getPath (key)	number	File	Gets the value of a key of type File.
getReference (key)	number	ActionReference	Gets the value of a key of type ActionReference.
getString (key)	number	string	Gets the value of a key of type string.
getType (key)	number	DescValueType	Gets the type of a key.
getUnitDoubleType (key)	number	number	Gets the unit type of a key of type UnitDouble.
getUnitDoubleValue (key)	number	number	Gets the value of a key of type UnitDouble.
hasKey (key)	number	boolean	Checks whether the descriptor contains the provided key.
isEqual (otherDesc)	ActionDescriptor	boolean	Determines whether the descriptor is the same as another descriptor.
putBoolean (key, value)	number boolean		Sets the value for a key whose type is boolean.
putClass (key, value)	number number		Sets the value for a key whose type is class.
putData (key, value)	number string		Puts raw byte data as a string value.
putDouble (key, value)	number number		Sets the value for a key whose type is double.
putEnumerated (key, enumType, value)	number number number		Sets the enumeration type and value for a key.

Method	Parameter type	Returns	What it does (Continued)
putInteger (key, value)	number number		Sets the value for a key whose type is integer.
putLargeInteger (key, value)	number number		Sets the value for a key whose type is large integer.
putList (key, value)	number ActionList		Sets the value for a key whose type is an ActionList object.
putObject (key, classID, value)	number number ActionDescriptor		Sets the value for a key whose type is an object, represented by an Action Descriptor.
putPath (key, value)	number File		Sets the value for a key whose type is path.
putReference (key, value)	number ActionReference		Sets the value for a key whose type is an object reference.
putString (key, value)	number string		Sets the value for a key whose type is string.
putUnitDouble (key, unitID, value)	number number number		Sets the value for a key whose type is a unit value formatted as a double.
toStream ()		string	Gets the entire descriptor as a stream of bytes, for writing to disk.

ActionList

This object provides an array-style mechanism for storing data. It can be used for low-level access into Photoshop.

This object is ideal when storing data of the same type. All items in the list must be of the same type.

You can use the "put" methods, such as putBoolean(), to append new elements, and can clear the entire list using clear(), but cannot otherwise modify the list.

Note: The `ActionList` object is part of the Action Manager functionality. For details on using the Action Manager, see the *Photoshop CS6 Scripting Guide*.

Properties

Property	Value type	What it is
count	number	Read-only. The number of commands that comprise the action.
typename	string	Read-only. The class name of the referenced `ActionList` object.

Methods

With the exception of the `clear()` method, you use the methods of this object to either get the value of a specific type of data in the list or set (put) the value type.

Method	Parameter type	Returns	What it does
clear ()			Clears the list.
getBoolean (index)	number	boolean	Gets the value of a list element of type boolean.
getClass (index)	number	number	Gets the value of a list element of type class.
getData (index)	number	string	Gets raw byte data as a string value.
getDouble (index)	number	number	Gets the value of a list element of type double.
getEnumerationType (index)	number	number	Gets the enumeration type of a list element.
getEnumerationValue (index)	number	number	Gets the enumeration value of a list element.
getInteger (index)	number	number	Gets the value of a list element of type integer.

Method	Parameter type	Returns	What it does (Continued)
getLargeInteger (index)	number	number	Gets the value of a list element of type large integer.
getList (index)	number	ActionList	Gets the value of a list element of type list.
getObjectType (index)	number	number	Gets the class ID of a list element of type object.
getObjectValue (index)	number	ActionDescriptor	Gets the value of a list element of type object.
getPath (index)	number	File	Gets the value of a list element of type File.
getReference (index)	number	ActionReference	Gets the value of a list element of type ActionReference.
getString (index)	number	string	Gets the value of a list element of type string.
getType (index)	number	DescValueType	Gets the type of a list element.
getUnitDoubleType (index)	number	number	Gets the unit value type of a list element of type Double.
getUnitDoubleValue (index)	number	number	Gets the unit value of a list element of type double.
putBoolean (value)	boolean		Appends a new value, true or false.
putClass (value)	number		Appends a new value, a class or data type.
putData (value)	string		Appends a new value, a string containing raw byte data.
putDouble (value)	number		Appends a new value, a double.
putEnumerated (enumType, value)	number number		Appends a new value, an enumerated (constant) value.
putInteger (value)	number		Appends a new value, an integer.
putLargeInteger (value)	number		Appends a new value, a large integer.
putList (value)	ActionList		Appends a new value, a nested action list.

Method	Parameter type	Returns	What it does (Continued)
putObject (classID, value)	number ActionDescriptor		Appends a new value, an object.
putPath (value)	File		Appends a new value, a path.
putReference (value)	ActionReference		Appends a new value, a reference to an object created in the script.
putString (value)	string		Appends a new value, a string.
putUnitDouble (classID, value)	number number		Appends a new value, a unit/value pair.

ActionReference

This object provides information about what the action is refering to. For example, when referring to the name of something you might use keyName. The reference would also need to know what name you are referring to. In this case you could use classDocument for the name of the document or classLayer for the name of the layer. It can be used for low-level access into Photoshop.Contains data associated with an ActionDescriptor.

Properties

Property	Value type	What it does
typename	string	Read-only. The class name of the referenced Action object.

Methods

Method	Parameter type	Returns	What it does
getContainer ()		ActionReference	Gets a reference contained in this reference. Container references provide additional pieces to the reference. This looks like another reference, but it is actually part of the same reference.
getDesiredClass ()		number	Gets a number representing the class of the object.
getEnumeratedType ()		number	Gets the enumeration type.
getEnumeratedValue ()		number	Gets the enumeration value.
getForm ()		ReferenceFormType	Gets the form of this action reference.
getIdentifier ()		number	Gets the identifier value for a reference whose form is identifier.
getIndex ()		number	Gets the index value for a reference in a list or array.
getName ()		string	Gets the name of a reference.
getOffset ()		number	Gets the offset of the object’s index value.
getProperty ()		number	Gets the property ID value.

Method	Parameter type	Returns	What it does (Continued)
putClass (desiredClass)	number		Puts a new class form and class type into the reference.
putEnumerated (desiredClass, enumType, value)	number number number		Puts an enumeration type and ID into a reference along with the desired class for the reference.
putIdentifier (desiredClass, value)	number number		Puts a new identifier and value into the reference.
putIndex (desiredClass, value)	number number		Puts a new index and value into the reference.
putName (desiredClass, value)	number string		Puts a new name and value into the reference.
putOffset (desiredClass, value)	number number		Puts a new offset and value into the reference.
putProperty (desiredClass, value)	number number		Puts a new property and value into the reference.

Application

The Adobe Adobe Photoshop CS6 application object, which is the root of the object model and provides access to all other objects. This object provides application-wide information, such as application defaults and available fonts. It provides many important methods , such as those for opening files and loading documents.

To access the properties and methods, you can use the pre-defined global variable `app`. For example:

```
var docRef = app.documents.add(800, 600, 72, "docRef", NewDocumentMode.RGB);
```

The properties and methods of the `Application` object are also available at the top level; you can omit references to the `Application` object altogether. For example:

```
var docRef = documents.add(800, 600, 72, "docRef", NewDocumentMode.RGB);
```

This usage can be somewhat ambiguous; for clarity, it is recommended that you use an explicit reference to `app`.

Properties

Property	Value type	What it is
activeDocument	Document	Read-write. The frontmost document. Setting this property is equivalent to clicking an open document in the Adobe Photoshop CS6 application to bring it to the front of the screen. **Tip:** If there is no open document, accessing this property throws an exception.
backgroundColor	SolidColor	Read-write. The default background color and color style for documents.
build	string	Read-only. Information about the application.
colorSettings	string	Read-write. The name of the current color settings, as selected with Edit > Color Settings.
displayDialogs	DialogModes	Read-write. The dialog mode for the application, which controls what types of dialogs should be displayed when running scripts.
documents	Documents	Read-only. The collection of open documents. This is the primary point of access for documents that are currently open in the application. The array allows you to access any open document, or to iterate through all open documents.
fonts	TextFonts	Read-only. The fonts installed on this system.
foregroundColor	SolidColor	Read-write. The default foreground color (used to paint, fill, and stroke selections).
freeMemory	number	Read-only. The amount of unused memory available to Adobe Photoshop CS6.

Property	Value type	What it is (Continued)
locale	string	Read-only. The language location of the application. An Adobe locale code consists of a 2-letter ISO-639 language code and an optional 2-letter ISO 3166 country code separated by an underscore. Case is significant. For example, en_US, en_UK, ja_JP, de_DE, fr_FR.
macintoshFileTypes	array of string	Read-only. A list of file image types Adobe Photoshop CS6 can open.
measurementLog	MeasurementLog	The log of measurements taken.
name	string	Read-only. The application's name.
notifiers	Notifiers	Read-only. The collection of notifiers currently configured (in the Scripts Events Manager menu in the Adobe Photoshop CS6 application).
notifiersEnabled	boolean	Read-write. True if all notifiers are enabled.
path	File	Read-only. The full path to the location of the Adobe Photoshop CS6 application.
playbackDisplayDialogs	DialogModes	Read-write. The dialog mode for playback mode, which controls what types of dialog to display when playing back a recorded action with the Actions palette.
playbackParameters	ActionDescriptor	Read-write. Stores and retrieves parameters used as part of a recorded action. Can be used, for example, to control playback speed.
preferences	Preferences	Read-only. The application preference settings (equivalent to selecting **Edit > Preferences** in the Adobe Photoshop CS6 application in Windows or Photoshop > Preferences in Mac OS).
preferencesFolder	File	Read-only. The full path to the Preferences folder.
recentFiles	array of File	Read-only. Files in the Recent Files list.
scriptingBuildDate	string	Read-only. The build date of the Scripting interface.
scriptingVersion	string	Read-only. The version of the Scripting interface.
systemInformation	string	Read-only. Runtime details of the application and system.
typename	string	Read-only. The class name of the referenced app object.

Property	Value type	What it is (Continued)
version	string	Read-only. The version of Adobe Photoshop application you are running.
windowsFileTypes	array of string	Read-only. A list of file image extensions Adobe Photoshop CS6 can open.

Methods

Method	Parameter type	Returns	What it does
batch (inputFiles, action, from [, options])	array of File string string BatchOptions	string	Runs the batch automation routine (similar to the **File > Automate > Batch** command). The inputFiles parameter specifies the sources for the files to be manipulated by the batch command.
beep ()			Causes a "beep" sound.
bringToFront ()			Makes Adobe Photoshop CS6 the active (front-most) application.
charIDToTypeID (charID)	string	number	Converts from a four character code (character ID) to a runtime ID.
doAction (action, from)	string string		Plays an action from the Actions palette. The action parameter is the name of the action, the from parameter is the name of the action set.
eraseCustomOptions (key)	string		Erases the user object with specified ID value from the Photoshop registry.
executeAction (eventID [, descriptor] [, displayDialogs])	number ActionDescriptor DialogModes	ActionDescriptor	Plays an Action Manager event.
executeActionGet (reference)	ActionReference	ActionDescriptor	Obtains information about a predefined or recorded action.
featureEnabled (name)	string	boolean	Determines whether the feature specified by name is enabled. The following features are supported as values for name: "photoshop/extended" "photoshop/standard" "photoshop/trial"

Method	Parameter type	Returns	What it does (Continued)
getCustomOptions (key)	string	ActionDescriptor	Retreives user objects in the Photoshop registry for the ID with value `key`.
isQuicktimeAvailable ()		boolean	Returns true if Quicktime is installed.
load (document)	File		Loads a support file (as opposed to a Photoshop image document) from the specified location.
makeContactSheet (inputFiles [, options])	array of File ContactSheetOptions	string	DEPRECATED for Adobe Photoshop CS4.
makePDFPresentation (inputFiles, outputFiles [, options])	array of File File PresentationOptions	string	DEPRECATED for Adobe Photoshop CS4.
makePhotoGallery (inputFolder, outputFolder [, options])	File File GalleryOptions	string	DEPRECATED for Adobe Photoshop CS4.
makePhotomerge (inputFiles)	array of File	string	DEPRECATED for Adobe Photoshop CS6. Use provided script: `runphotomergeFromScript = true;` `$.evalFile( app.path + "Presets/Scripts/Photomerge.jsx")` `photomerge.createPanorama( fileList, displayDialog );` Merges multiple files into one, with user interaction required.
makePicturePackage (inputFiles [, options])	array of File PicturePackageOptions	string	DEPRECATED for Adobe Photoshop CS4.

Method	Parameter type	Returns	What it does (Continued)
open (document [, as] [, asSmartObject])	File object or OpenDocumentType boolean	Document	Opens the specified document. Use the optional as parameter to specify the file format using the constants in OpenDocumentType; or, you can specify a file format together with its open options using these objects: CameraRAWOpenOptions DICOMOpenOptions EPSOpenOptions PDFOpenOptions PhotoCDOpenOptions RawFormatOpenOptions Use the optional parameter asSmartObject (default: false) to create a smart object around the opened document. See the Application sample scripts for an example of using the File object in the open method.
openDialog ()		array of File	Invokes the Photoshop Open dialog box for the user to select files. Returns an array of File objects for the files selected in the dialog.
purge (target)	PurgeTarget		Purges one or more caches.
putCustomOptions (key, customObject [, persistent])	string ActionDescriptor boolean		Saves a customized settings object in the Photoshop registry. key is the unique identifier for your custom settings. customObject is the object to save in the registry. persistent indicates whether the object should persist once the script has finished.
refresh ()			Pauses the script while the application refreshes. Use to slow down execution and show the results to the user as the script runs. Use carefully; your script runs much more slowly when using this method.
refreshFonts ()			Force the font list to get updated.
runMenuItem (menuID)	number		Run a menu item given the menu ID.

Method	Parameter type	Returns	What it does (Continued)
showColorPicker ()		boolean	Returns false if dialog is cancelled, true otherwise.
stringIDToTypeID (stringID)	string	number	Converts from a string ID to a runtime ID.
togglePalettes ()			Toggle palette visibility.
typeIDToCharID (typeID)	number	string	Converts from a runtime ID to a character ID.
typeIDToStringID (typeID)	number	string	Converts from a runtime ID to a string ID.

Application sample scripts

Application.jsx

This script invokes an alert box to display Properties important to an application such as version number, the path to the application, the amount of memory available, and the number of documents open.

When a user presses the OK button on the alert box, a second dialog opens, which asks users whether they would like the foreground and background colors set for the document presently open. If no document is open, the script opens a new document for the user.

The script (with no document open) produces a progression of three dialogs.

```
//Create a Welcome message
// Use the name and version properties of the application object to
// Append the application’s name and version to the Welcome message
// use "\r" to insert a carriage return
// use the combination operator += to append info to the message
var message = "Welcome to " + app.name
message += " version " + app.version + "\r\r"

// find out where Adobe Photoshop CS6 is installed
// and add the path to the message
// add the optional parameter fsName to the path property
// to display the file system name in the most common format
message += "I’m installed in " + app.path.fsName + "\r\r"

// see how much memory Adobe Photoshop CS6 has to play with
message += "You have this much memory available for Adobe Photoshop CS6: " +
app.freeMemory + "\r\r"

// use the length property of the documents object to
// see how many documents are open
var documentsOpen = app.documents.length
message += "You currently have " + documentsOpen + " document(s) open.\r\r"

// display the message to the user
alert(message)

// answer will be true for a "Yes" answer and false for a "No" answer
var answer = confirm("Set the foreground and background to my favorite colors?")
```

```
// set the colors
if (answer) {
   // I don't have a favorite color. Why did I ask you may wonder?
   app.foregroundColor.rgb.red = Math.random() * 255
   app.foregroundColor.rgb.green = Math.random() * 255
   app.foregroundColor.rgb.blue = Math.random() * 255
   app.backgroundColor.rgb.red = Math.random() * 255
   app.backgroundColor.rgb.green = Math.random() * 255
   app.backgroundColor.rgb.blue = Math.random() * 255
}

// Open a document
if (app.documents.length == 0) {

   // use the application's path and the offset to the samples folder
   var sampleDocToOpen = File(app.path + "/Samples/Fish.psd")

   // compose a message with the name of the file
   message = "Would you like me to open a sample for you? ("
   message += sampleDocToOpen.fsName
   message += ")"

   // ask the user another question
   answer = confirm(message)

   // open the document accordingly
   if (answer) {
   open(sampleDocToOpen)
   }
}
```

PDFPresentation.jsx

This script presents a progression of images as an Adobe PDF slide show.

```
// use all the files in the Samples folder
var inputFolder = new Folder(app.path + "/Samples/")

// see if we have something interesting
if (inputFolder != null) {

    // get all the files found in this folder that are Adobe Photoshop CS6 (.psd
format)
    var inputFiles = inputFolder.getFiles("*.psd")

    // output to the desktop
    var outputFile = File("~/Desktop/JavaScriptPresentation.pdf")

    // there are defaults but I like to set the options myself
    var options = new PresentationOptions
    options.presentation = true
    options.view = true
    options.autoAdvance = true
    options.interval = 5
    options.loop = true
    options.transition = TransitionType.RANDOM

    // create the presentation
    makePDFPresentation(inputFiles, outputFile, options)
```

```
alert("Presentation file saved to: " + outputFile.fsName)
}
```

ArtLayer

An object within a document that contains the visual elements of the image (equivalent to a layer in the Adobe Photoshop CS6 application).

Access an art layer in a document through the Document.artLayers collection. You can access a layer by name; for example:

```
var layerRef = app.activeDocument.artLayers.getByName("my layer");
layerRef.allLocked = true;
```

Access the art layers in a layer set through the LayerSet.artLayers collection in the parent set.

Properties

Property	Value type	What it is
allLocked	boolean	Read-write. True to completely lock the contents and settings of this layer.
blendMode	BlendMode	Read-write. The blending mode.
bounds	array of UnitValue	Read-only. An array of coordinates that describes the bounding rectangle of the layer.
fillOpacity	number [0.0..100]	Read-write. The interior opacity of the layer, a percentage value.
filterMaskDensity	double	Read-write. The density of the filter mask (between 0.0 and 250.0)
filterMaskFeather	double	Read-write. The feather of the filter mask (between 0.0 and 250.0)
grouped	boolean	Read-write. True if this layer is grouped with the layer beneath it.
isBackgroundLayer	boolean	Read-write. True if this is the background layer of the document. A document can have only one background layer. If there is no background layer, setting this to true causes this to become the background layer.
kind	LayerKind	Read-write. Sets the type (such as 'text layer') for an empty layer. Valid only when the layer is empty and when `isBackgroundLayer` is `false`. See isBackgroundLayer. You can use the `kind` property to make a background layer a normal layer; however, to make a layer a background layer, you must set `isBackgroundLayer` to `true`.
layerMaskDensity	double	Read-write. The density of the layer mask (between 0.0 and 100.0)

Property	Value type	What it is (Continued)
layerMaskFeather	double	Read-write. The feather of the layer mask (between 0.0 and 250.0)
linkedLayers	array of ArtLayer or LayerSet	Read-only. The layers linked to this layer. See ArtLayer.link.
name	string	Read-write. The name.
opacity	number [0.0..100.0].	Read-write. The master opacity of the layer, a percentage value.
parent	Document	Read-only. The object's container.
pixelsLocked	boolean	Read-write. True if the pixels in the layer's image cannot be edited using the paintbrush tool.
positionLocked	boolean	Read-write. True if the pixels in the layer's image cannot be moved within the layer.
textItem	TextItem	Read-only. The text item that is associated with the layer. Valid only when kind = LayerKind.TEXT.
transparentPixelsLocked	boolean	Read-write. True if editing is confined to the opaque portions of the layer.
typename	string	Read-only. The class name of the referenced artLayer object.
vectorMaskDensity	double	Read-write. The density of the vector mask (between 0.0 and 250.0)
vectorMaskFeather	double	Read-write. The feather of the vector mask (between 0.0 and 250.0)
visible	boolean	Read-write. True if the layer is visible.
xmpMetadata	xmpMetadata	Read-write. Metadata for the layer.

Methods

Method	Parameter type	Returns	What it does
adjustBrightnessContrast (brightness, contrast)	number number		Adjusts the brightness in the range [-100..100] and contrast [-100..100].
adjustColorBalance ([shadows] [, midtones] [, highlights] [, preserveLuminosity]	array of number array of number array of number boolean		Adjusts the color balance of the layer's component channels. For shadows, midtones, and highlights, the array must include three values in the range [-100..100], which represent cyan or red, magenta or green, and yellow or blue, when the document mode is CMYK or RGB. See Document.mode.
adjustCurves (curveShape)	array of array of number		Adjusts the tonal range of the selected channel using up to fourteen points. Each value in the curveShape array is a point pair, an array of an x and y integer value.
adjustLevels (inputRangeStart, inputRangeEnd, inputRangeGamma, outputRangeStart, outputRangeEnd)	number [0..253] number [(*start* + 2)..255] number [0.10..9.99] number [0..253] number [(*start* + 2)..255]		Adjusts the levels of the selected channels
applyAddNoise (amount, distribution, monochromatic)	number [0.1..400] NoiseDistribution boolean		Applies the Add Noise filter amount is a percentage value.
applyAverage ()			Applies the Average filter.
applyBlur ()			Applies the Blur filter.
applyBlurMore ()			Applies the Blur More filter.
applyClouds ()			Applies the Clouds filter.
applyCustomFilter (characteristics, scale, offset)	array of number number number		Applies a custom filter. The characteristics array has 25 members. See Adobe Photoshop CS6 Help for specific instructions.

Method	Parameter type	Returns	What it does (Continued)
applyDeInterlace (eliminateFields, createFields)	EliminateFields CreateFields		Applies the De-Interlace filter.
applyDespeckle ()			Applies the Despeckle filter.
applyDifferenceClouds ()			Applies the Difference Clouds filter.
applyDiffuseGlow (graininess, glowAmount, clearAmount)	number [0..10] number [0..20] number [0..20]		Applies the Diffuse Glow filter.
applyDisplace (horizontalScale, verticalScale, displacement, undefinedareas, displacementMapFiles)	number [-999..999] number [-999..999] DisplacementMapType UndefinedAreas File		Applies the Displace filter using the specified horizontal and vertical scale, mapping type, treatment of undistorted areas, and path to the distortion image map.
applyDustAndScratches (radius, threshold)	number [1..100] number [0..255]		Applies the Dust & Scratches filter.
applyGaussianBlur (radius)	number [0.1..250.0]		Applies the Gaussian Blur filter within the specified radius (in pixels)
applyGlassEffect (distortion, smoothness, scaling [, invert] [, texture] [, textureFile])	number [0..20] number [1..15] number [50..200] boolean TextureType File		Applies the Glass filter. scaling is a percentage value.
applyHighPass (radius)	number [0.1..250.0]		Applies the High Pass filter within the specified radius.

Method	Parameter type	Returns	What it does (Continued)
applyLensBlur ([source] [, focalDistance] [, invertDepthMap] [, shape] [, radius] [, bladeCurvature] [, rotation] [, brightness] [, threshold] [, amount] [, distribution] [, monochromatic])	DepthMapSource number boolean Geometry number number number number number number NoiseDistribution boolean		Applies the Lens Blur filter. source: The source for the depth map (default: DepthMapSource.NONE) focalDistance : The blur focal distance for the depth map (default: 0). invertDepthMask : True if the depth map is inverted (default: false). shape: The shape of the iris (default: Geometry.HEXAGON) radius: The radius of the iris (default: 15). bladeCurvature: The blade curvature of the iris (default: 0). rotation: The rotation of the iris (default: 0) brightness: The brightness for the specular highlights (default: 0). threshold: The threshold for the specular highlights (default: 0). amount: The amount of noise (default: 0) distribution: The distribution value for the noise (default: NoiseDistribution.UNIFORM). monochromatic: True if the noise is monochromatic (default: false).
applyLensFlare (brightness, flareCenter, lensType)	number array(UnitValue) LensType		Applies the Lens Flare filter with the specified brightness (0 - 300, as a percentage), the x and y coordinates (unit value) of the flare center, and the lens type.
applyMaximum (radius)	number [1..100]		Applies the Maximum filter within the specified radius (in pixels).
applyMedianNoise (radius)	number [1..100]		Applies the Median Noise filter within the specified radius (in pixels).

Method	Parameter type	Returns	What it does (Continued)
applyMinimum (radius)	number [1..100]		Applies the Minimum filter within the specified radius (in pixels) (1 - 100).
applyMotionBlur (angle, radius)	number [-360..360] number [1..999]		Applies the Motion Blur filter.
applyNTSC ()			Applies the NTSC colors filter.
applyOceanRipple (size, magnitude)	number [1..15] number [0..20]		Applies the Ocean Ripple filter.
applyOffset (horizontal, vertical, undefinedAreas)	UnitValue UnitValue OffsetUndefinedAreas		Moves the layer the specified amount horizontally and vertically (min/max amounts depend on layer size), leaving an undefined area at the layer's original location.
applyPinch (amount)	number [-100..100]		Applies the Pinch filter. `amount` is a percentage value.
applyPolarCoordinates (conversion)	PolarConversionType		Applies the Polar Coordinates filter.
applyRadialBlur (amount, blurMethod, blurQuality)	number [1..100] RadialBlurMethod RadialBlurQuality		Applies the Radial Blur filter in the specified amount, using either a spin or zoom effect and the specified quality.
applyRipple (amount, size)	number [-999..999] RippleSize		Applies the Ripple filter in the specified amount, throughout the image and in the specified size.
applySharpen ()			Applies the Sharpen filter.
applySharpenEdges ()			Applies the Sharpen Edges filter.
applySharpenMore ()			Applies the Sharpen More filter.
applyShear (curve, undefinedAreas)	array of array of number UndefinedAreas		Applies the Shear filter. The `curve` defines a curve with [2..255] points. Each value in the `curve` array is a point pair, an array of an x and y integer value.

Method	Parameter type	Returns	What it does (Continued)
applySmartBlur (radius, threshold, blurQuality, mode)	number [0.1..100.0] number [0.1..100.0] SmartBlurQuality SmartBlurMode		Applies the Smart Blur filter.
applySpherize (amount, mode)	number [-100..100] SpherizeMode		Applies the Spherize filter. amount is a percentage value.
applyStyle (styleName)	string		Applies the specified style to the layer. You must use a style from the Styles list in the Layer Styles Palette.
applyTextureFill (textureFile)	File		Applies the Texture Fill filter.
applyTwirl (angle)	number [-999..999]		Applies the Twirl filter.
applyUnSharpMask (amount, radius, threshold)	number [1..500] number [0.1..250.0] number [0..255]		Applies the Unsharp Mask filter. (amount is a percentage value.
applyWave (generatorNumber, minimumWavelength, maximumWavelength, minimumAmplitude, maximumAmplitude, horizontalScale, verticalScale, waveType, undefinedAreas, randomSeed)	number [1..999] number [1..998] number [2..*min*+1] number [1..998] number [2..*min*+1] number [1..100] number [1..100] WaveType UndefinedAreas number		Applies the Wave filter. Scale factors are percentage values.
applyZigZag (amount, ridges, style)	number [-100..100] number [0..20] ZigZagType		Applies the Zigzag filter.
autoContrast ()			Adjusts the contrast of the selected channels automatically.
autoLevels ()			Adjusts the levels of the selected channels using the auto levels option.
clear ()			Cuts the layer without moving it to the clipboard.

Method	Parameter type	Returns	What it does (Continued)
copy ([merge])	boolean		Copies the layer to the clipboard. When the optional argument is set to `true`, a merged copy is performed (that is, all visible layers are copied to the clipboard).
cut ()			Cuts the layer to the clipboard.
desaturate ()			Converts a color image to a grayscale image in the current color mode by assigning equal values of each component color to each pixel.
duplicate ([relativeObject] [, insertionLocation])	ArtLayer or LayerSet ElementPlacement	ArtLayer or LayerSet	Creates a duplicate of the object on the screen.
equalize ()			Redistributes the brightness values of pixels in an image to more evenly represent the entire range of brightness levels within the image.
invert ()			Inverts the colors in the layer by converting the brightness value of each pixel in the channels to the inverse value on the 256-step color-values scale.
link (with)	ArtLayer or LayerSet		Links the layer with the specified layer.
merge ()		ArtLayer	Merges the layer down, removing the layer from the document; returns a reference to the art layer that this layer is merged into.

Method	Parameter type	Returns	What it does (Continued)
mixChannels (outputChannels [, monochrome])	array of array of number boolean		Modifies a targeted (output) color channel using a mix of the existing color channels in the image. The outputChannels parameter is an array of channel specifications. For each component channel, specify a list of adjustment values in the range [-200..200] followed by a 'constant' value [-200..200].) When monochrome = true, the maximum number of channel value specifications is 1. Valid only when docRef.mode = DocumentMode.RGB or CMYK. RGB arrays must include four values. CMYK arrays must include five values.
move (relativeObject, insertionLocation)	ArtLayer or LayerSet ElementPlacement		Moves the layer relative to the object specified in parameters. For art layers, only the constant values ElementPlacement. PLACEBEFORE and PLACEAFTER are valid. For layer sets, only the constant values ElementPlacement. PLACEBEFORE and INSIDE are valid.
photoFilter ([fillColor] [, density] [, preserveLuminosity])	SolidColor number [1..100] boolean		Adjust the layer's color balance and temperature as if a color filter had been applied. density is a percentage value.
posterize (levels)	number [2..225]		Specifies the number of tonal levels for each channel and then maps pixels to the closest matching level.
rasterize (target)	RasterizeType		Converts the targeted contents in the layer into a flat, raster image.
remove ()			Deletes the object.

Method	Parameter type	Returns	What it does (Continued)
resize ([horizontal] [, vertical] [, anchor])	number number AnchorPosition		Resizes the layer to the specified dimensions (as a percentage of its current size) and places it in the specified position.
rotate (angle [, anchor])	number AnchorPosition		Rotates rotates the layer around the specified anchor point (default: MIDDLECENTER).
selectiveColor (selectionMethod [, reds] [, yellows] [, greens] [, cyans] [, blues] [, magentas] [, whites] [, neutrals] [, blacks])	AdjustmentReference array of number array of number array of number array of number array of number array of number array of number array of number array of number		Modifies the amount of a process color in a specified primary color without affecting the other primary colors. Each color array must have four values.
shadowHighlight ([shadowAmount] [, shadowWidth] [, shadowRadius] [, highlightAmount] [, highlightWidth] [, highlightRadius] [, colorCorrection] [, midtoneContrast] [, blackClip] [, whiteClip])	number [0..100] number [0.100] number [0..2500] number [0..100] number [0..100] number [0..2500] number [-100..100] number [-100..100] number [0.000..50.000] number [0.000..50.000]		Adjusts the range of tones in the image's shadows and highlights. Amounts and widths are percentage values. Radius values are in pixels.
threshold (level)	number [1..255]		Converts grayscale or color images to high-contrast, B/W images by converting pixels lighter than the specified threshold to white and pixels darker than the threshold to black.
translate ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the layer the specified amount (in the given unit) relative to its current position.
unlink ()			Unlinks the layer.

ArtLayer sample Script

The following script opens all the files in the samples folder, creating one multi-layered document. Each layer is pasted into one of four quadrants and given 50% transparency. Finally the layers are sorted by name.

ArtLayer.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

//Close all the open documents
 while (app.documents.length) {
   app.activeDocument.close()
}

// Create a new document to merge all the samples into
var mergedDoc = app.documents.add(1000, 1000, 72, "Merged Samples",
NewDocumentMode.RGB, DocumentFill.TRANSPARENT, 1)

// Use the path to the application and append the samples folder
var samplesFolder = Folder(app.path + "/Samples/")

//Get all the files in the folder
var fileList = samplesFolder.getFiles()

// open each file
for (var i = 0; i < fileList.length; i++) {
   // The fileList is folders and files so open only files
   if (fileList[i] instanceof File) {
          open(fileList[i])

          // use the document name for the layer name in the merged document
          var docName = app.activeDocument.name

          // flatten the document so we get everything and then copy
          app.activeDocument.flatten()
          app.activeDocument.selection.selectAll()
          app.activeDocument.selection.copy()

          // don't save anything we did
          app.activeDocument.close(SaveOptions.DONOTSAVECHANGES)

          // make a random selection on the document to paste into
          // by dividing the document up in 4 quadrants and pasting
          // into one of them by selecting that area
          var topLeftH = Math.floor(Math.random() * 2)
          var topLeftV = Math.floor(Math.random() * 2)
          var docH = app.activeDocument.width.value / 2
          var docV = app.activeDocument.height.value / 2
          var selRegion = Array(Array(topLeftH * docH, topLeftV * docV),
             Array(topLeftH * docH + docH, topLeftV * docV),
             Array(topLeftH * docH + docH, topLeftV * docV + docV),
             Array(topLeftH * docH, topLeftV * docV + docV),
             Array(topLeftH * docH, topLeftV * docV))
          app.activeDocument.selection.select(selRegion)
          app.activeDocument.paste()

          // change the layer name and opacity
          app.activeDocument.activeLayer.name = docName
          app.activeDocument.activeLayer.fillOpacity = 50
```

```
   }
}

// sort the layers by name
for (var x = 0; x < app.activeDocument.layers.length; x++) {
   for (var y = 0; y < app.activeDocument.layers.length - 1 - x; y++) {
          // Compare in a non-case sensitive way
          var doc1 = app.activeDocument.layers[y].name
          var doc2 = app.activeDocument.layers[y + 1].name
          if (doc1.toUpperCase() > doc2.toUpperCase()) {
             app.activeDocument.layers[y].move(app.activeDocument.layers[y+1],
                ElementPlacement.PLACEAFTER)
      }
   }
}

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs
```

ArtLayers

The collection of ArtLayer objects in a document or layer set.

Access through the Document.artLayers or LayerSet.artLayers collection. For example:

```
var layerRef = docRef.artLayers.add()
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `artLayers` collection.
parent	Document	Read-only. The object's container.
typename	string	Read-only. The class name of the referenced `artLayers` object.

Methods

Method	Parameter type	Returns	What it does
add ()		ArtLayer	Creates a new art layer in the document and adds the new object to this collection.
getByName (name)	string	ArtLayer	Get the first element in the `artLayers` collection with the provided name.
removeAll ()			Removes all elements from the `artLayers` collection.

BatchOptions

Options for running a batch operation using the Application.batch() method.

JavaScript only supports folders as sources for batch commands. Specify the batch source folder as the `inputFiles` parameter of the Application.batch() method.

Properties

Property	Value type	What it is
destination	`BatchDestinationType`	Read-write. The type of destination for the processed files (default: `BatchDestinationType.NODESTINATION`).
destinationFolder	`Folder`	Read-write. The folder location for the processed files. Valid only when destination = `BatchDestinationType.FOLDER`.
errorFile	`File`	Read-write. The file in which to log errors encountered. To display errors on the screen (and stop batch processing when errors occur) leave blank.
fileNaming	`array of FileNamingType`	Read-write. A list of file naming options (maximum: 6). Valid only when destination = `BatchDestinationType.FOLDER`.
macintoshCompatible	`boolean`	Read-write. True to make the final file names Macintosh compatible (default: `true`). Valid only when destination = `BatchDestinationType.FOLDER`.
overrideOpen	`boolean`	Read-write. True to override action open commands (default: `false`).
overrideSave	`boolean`	Read-write. True to override save as action steps with the specified destination (default: `false`). Valid only when destination = `BatchDestinationType.FOLDER` or `SAVEANDCLOSE`.
startingSerial	`number`	Read-write. The starting serial number to use in naming files (default: `1`). Valid only when destination = `BatchDestinationType.FOLDER`.
suppressOpen	`boolean`	Read-write. True to suppress the file open options dialogs (default: `false`).

Property	Value type	What it is (Continued)
`suppressProfile`	`boolean`	Read-write. True to suppress the color profile warnings (default: `false`).
`typename`	`string`	Read-only. The class name of the referenced `batchOptions` object.
`unixCompatible`	`boolean`	Read-write. True to make the final file name Unix compatible (default: `true`). Valid only when destination = `BatchDestinationType.FOLDER`.
`windowsCompatible`	`boolean`	Read-write. True to make the final file names Windows compatible (default: `true`). Valid only when destination = `BatchDestinationType.FOLDER`.

BitmapConversionOptions

Options for converting an image to bitmap mode, using Document.changeMode() with `ChangeMode.Bitmap`.

Convert color images to grayscale before converting the image to bitmap mode. See the ArtLayer.desaturate() method.

Properties

Property	Value type	What it is
`angle`	`number [-180..180]`	Read-write. The angle (in degrees) at which to orient individual dots. See shape. Valid only when method = `BitmapConversionType.HALFTONESCREEN`.
`frequency`	`number [1.0..999.99]`	Read-write. The number of printer dots (per inch) to use. Valid only when method = `BitmapConversionType.HALFTONESCREEN`.
`method`	`BitmapConversionType`	Read-write. The conversion method to use (default: `BitmapConversionType.DIFFUSIONDITHER`).
`patternName`	`string`	Read-write. The name of the pattern to use. For information about pre-installed valid patterns, see Adobe Photoshop CS6 Help on the bitmap conversion command, or view the options availabe in the Custom Color drop down box after choosing the bitmap conversion command. Valid only when method = `BitmapConversionType.CUSTOMPATTERN`.
`resolution`	`number`	Read-write. The output resolution in pixels per inch (default: `72.0`).
`shape`	`BitmapHalfToneType`	Read-write. The dot shape to use. Valid only when method = `BitmapConversionType.HALFTONESCREEN`.
`typename`	`string`	Read-only. The class name of the referenced `bitmapConversionOptions` object.

BMPSaveOptions

Options for saving a document in BMP format using the Document.saveAs() method.

Properties

Property	Value type	What it is
`alphaChannels`	`boolean`	Read-write. True to save the alpha channels.
`depth`	`BMPDepthType`	Read-write. The number of bits per channel.
`flipRowOrder`	`boolean`	Read-write. True to write the image from top to bottom (default: `false`). Available only when osType = `OperatingSystem.WINDOWS`.
`osType`	`OperatingSystem`	Read-write. The target OS. (default: `OperatingSystem.WINDOWS`).
`rleCompression`	`boolean`	Read-write. True to use RLE compression. Available only when osType = `OperatingSystem.WINDOWS`.
`typename`	`string`	Read-only. The class name of the referenced `BMPSaveOptions` object.

CameraRAWOpenOptions

Options for opening a document in Camera RAW format using the Application.open() method.

Properties

Property	Value type	What it is
bitsPerChannel	BitsPerChannelType	Read-write. The number of bits per channel.
blueHue	number [-100..100]	Read-write. The blue hue of the shot.
blueSaturation	number [-100..100]	Read-write. The blue saturation of the shot.
brightness	number [0..150]	Read-write. The brightness of the shot.
chromaticAberrationBY	number [-100..100]	Read-write. The chromatic aberration B/Y of the shot.
chromaticAberrationRC	number [-100..100]	Read-write. The chromatic aberration R/C of the shot
colorNoiseReduction	number [0..100]	Read-write. The color noise reduction of the shot.
colorSpace	ColorSpaceType	Read-write. The colorspace for the image.
contrast	number [-50..100]	Read-write. The contrast of the shot.
exposure	number [-4.0..4.0]	Read-write. The exposure of the shot.
greenHue	number [-100..100]	Read-write. The green hue of the shot.
greenSaturation	number [-100..100]	Read-write. The green saturation of the shot.
luminanceSmoothing	number [0..100]	Read-write. The luminance smoothing of the shot.
redHue	number [-100..100]	Read-write. The red hue of the shot.
redSaturation	number [-100..100]	Read-write. The red saturation of the shot.
resolution	number [1..999]	Read-write. The resolution of the document in pixels per inch.
saturation	number [-100..100]	Read-write. The saturation of the shot.
settings	CameraRAWSettingsType	Read-write. The global settings for all Camera RAW options. Default: CameraRAWSettingsType.CAMERA.
shadows	number [0..100]	Read-write. The shadows of the shot.
shadowTint	number [-100..100]	Read-write. The shadow tint of the shot.
sharpness	number [0..100]	Read-write. The sharpness of the shot.
size	CameraRAWSize	Read-write. The size of the new document.
temperature	number [2000..50000]	Read-write. The temperature of the shot.

Property	Value type	What it is (Continued)
`tint`	`number [-150..150]`	Read-write. The tint of the shot.
`typename`	`string`	Read-only. The class name of the referenced `cameraRAWOpenOptions` object.
`vignettingAmount`	`number [-100..100]`	Read-write. The vignetting amount of the shot.
`vignettingMidpoint`	`number [-100..100]`	Read-write. The vignetting mid point of the shot.
`whiteBalance`	`WhiteBalanceType`	Read-write. The white balance options for the image. These are lighting conditions that affect color balance.

Channel

Information about a color element in the image.

Access through the Document.channels collection. You can access an individual channel object in this list by index or by name. For example, this accesses a channel object in the active document by name and assigns an `opacity` value:

```
var channelRef = app.activeDocument.channels.getByName("my channel");
channelRef.opacity = 22;
```

A channel is analogous to a plate in the printing process that applies a single color. The document's color mode determines the number of default channels; for example, an RGB document has three channels, red, green, and blue. A color can also have an alpha channel, which stores selections as masks, or a spot channel, which stores spot colors.

Properties

Property	Value type	What it is
color	SolidColor	Read-write. The color of the channel. Not valid when kind = ChannelType.COMPONENT.
histogram	array of number	Read-only. A histogram of the color of the channel. The array contains 256 members. Not valid when kind = ChannelType.COMPONENT. For component channel histogram values, use the histogram property of the Document object instead.
kind	ChannelType	Read-write. The type of the channel.
name	string	Read-write. The name of the channel.
opacity	number [0..100]	Read-write. The opacity to use for alpha channels or the solidity to use for spot channels. Valid only when kind = ChannelType.MASKEDAREA or SELECTEDAREA.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced channel object.
visible	boolean	Read-write. True if the channel is visible.

Methods

Method	Parameter type	Returns	What it does
duplicate ([targetDocument])	Document	Channel	Duplicates the channel.

Method	Parameter type	Returns	What it does
merge ()			Merges a spot channel into the component channels.
remove ()			Deletes the channel.

Channels

The collection of Channel objects in a document.

Access through the Document.channels collection property. For example:

```
var channelRef = app.activeDocument.channels.add()
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `channels` collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `channels` object.

Methods

Method	Parameter type	Returns	What it does
add ()		Channel	Creates a new channel object and adds it to this collection.
getByName (name)	string	Channel	Get the first element in the `channels` collection with the provided name.
removeAll ()			Removes all alpha channel objects from the `channels` collection.

Channels sample script

The following script opens a file if one is not already open, and then writes a histogram report (histogram.log) for the channels in the active document.

Note: This script contains a switch construction that uses a `break` statement. The `break` statement requires an ending semicolon (;), as in the following sample:

```
break;
```

Histogram.jsx

```
// Function to activate all the channels according to the documents mode
// Takes a document reference for input
function TurnOnDocumentHistogramChannels(inDocument) {

   // see how many channels we need to activate
   var visibleChannelCount = 0

   // based on the mode of the document
   switch (inDocument.mode) {

          case DocumentMode.BITMAP:
```

```
            case DocumentMode.GRAYSCALE:
            case DocumentMode.INDEXEDCOLOR:
                visibleChannelCount = 1
                break;

            case DocumentMode.DUOTONE:
                visibleChannelCount = 2
                break;

            case DocumentMode.RGB:
            case DocumentMode.LAB:
                visibleChannelCount = 3
                break;

            case DocumentMode.CMYK:
                visibleChannelCount = 4
                break;

            case DocumentMode.MULTICHANNEL:
            default:
                visibleChannelCount = inDocument.channels.length + 1
                break;
    }

    // now get the channels to activate into a local array
    var aChannelArray = new Array()

    // index for the active channels array
    var aChannelIndex = 0

    for(var channelIndex = 0; channelIndex < inDocument.channels.length;
                channelIndex++) {
            if (channelIndex < visibleChannelCount) {
                aChannelArray[aChannelIndex++] = inDocument.channels[channelIndex]
        }
    }

    // now activate them
    inDocument.activeChannels = aChannelArray

}

// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// if there are no documents open then try to open a sample file
if (app.documents.length == 0) {
    open(File(app.path + "/Samples/Fish.psd"))
}

// get a reference to the working document
var docRef = app.activeDocument
```

```
// create the output file
// first figure out which kind of line feeds we need
if ($.os.search(/windows/i) != -1) {
   fileLineFeed = "Windows"
} else {
   fileLineFeed = "Macintosh"
}

// create the output file accordingly
fileOut = new File("~/Desktop/Histogram.log")
fileOut.lineFeed = fileLineFeed
fileOut.open("w", "TEXT", "????")

// write out a header
fileOut.write("Histogram report for " + docRef.name)

// find out how many pixels I have
var totalCount = docRef.width.value * docRef.height.value

// more info to the out file
fileOut.write(" with a total pixel count of " + totalCount + "\n")

// channel indexer
var channelIndex = 0

// remember which channels are currently active
var myActiveChannels = app.activeDocument.activeChannels

// document histogram only works in these modes
if (docRef.mode == DocumentMode.RGB ||
    docRef.mode == DocumentMode.INDEXEDCOLOR ||
    docRef.mode == DocumentMode.CMYK) {

   // activate the main channels so we can get the documents histogram
   TurnOnDocumentHistogramChannels(docRef)

   // Output the documents histogram
   OutputHistogram(docRef.histogram, "Luminosity", fileOut)
}

// local reference to work from
var myChannels = docRef.channels

// loop through each channel and output the histogram
for (var channelIndex = 0; channelIndex < myChannels.length; channelIndex++) {

   // the channel has to be visible to get a histogram
   myChannels[channelIndex].visible= true

   // turn off all the other channels
   for (var secondaryIndex = 0; secondaryIndex < myChannels.length;
            secondaryIndex++) {
          if (channelIndex != secondaryIndex) {
             myChannels[secondaryIndex].visible= false
      }
   }

   // Use the function to dump the histogram
   OutputHistogram(myChannels[channelIndex].histogram,
          myChannels[channelIndex].name, fileOut)
```

```
}

// close down the output file
fileOut.close()
alert("Histogram file saved to: " + fileOut.fsName)

// reset the active channels
docRef.activeChannels = myActiveChannels

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs


// Utility function that takes a histogram and name
// and dumps to the output file
function OutputHistogram(inHistogram, inHistogramName, inOutFile) {

   // find ouch which count has the largest number
   // I scale everything to this number for the output
   var largestCount = 0

   // a simple indexer I can reuse
   var histogramIndex = 0

   // see how many samples we have total
   var histogramCount = 0

   // search through all and find the largest single item
   for (histogramIndex = 0; histogramIndex < inHistogram.length;
            histogramIndex++) {
         histogramCount += inHistogram[histogramIndex]
         if (inHistogram[histogramIndex] > largestCount)
            largestCount = inHistogram[histogramIndex]
   }

   // These should match
   if (histogramCount != totalCount) {
         alert("Something bad is happening!")
   }

   // see how much each "X" is going to count as
   var pixelsPerX = largestCount / 100

   // output this data to the file
   inOutFile.write("One X = " + pixelsPerX + " pixels.\n")

   // output the name of this histogram
   inOutFile.write(inHistogramName + "\n")

   // loop through all the items and output in the following format
   // 001
   // 002
   for (histogramIndex = 0; histogramIndex < inHistogram.length;
            histogramIndex++) {

         // I need an extra "0" for this line item to keep everything in line
         if (histogramIndex < 10)
            inOutFile.write("0")
```

```
            // I need an extra "0" for this line item to keep everything in line
            if (histogramIndex < 100)
                inOutFile.write("0")

            // output the index to file
            inOutFile.write(histogramIndex)

            // some spacing to make it look nice
            inOutFile.write(" ")

            // figure out how many X’s I need
            var outputX = inHistogram[histogramIndex] / largestCount * 100

            // output the X’s
            for (var a = 0; a < outputX; a++)
                inOutFile.write("X")

            inOutFile.write("\n")
    }

    inOutFile.write("\n")
}
```

CMYKColor

Defines a CMYK color, used in the SolidColor object.

See also GrayColor, HSBColor, LabColor, NoColor, RGBColor

Properties

Property	Value type	What it is
black	number [0.0..100.00]	Read-write. The black color value (as percent).
cyan	number [0.0..100.00]	Read-write. The cyan color value (as percent).
magenta	number [0.0..100.00]	Read-write. The magenta color value (as percent).
typename	string	Read-only. The class name of the referenced CMYKColor object.
yellow	number [0.0..100.00]	Read-write. The yellow color value (as percent).

ColorSampler

A color sampler for a document. Access through the Document.colorSamplers collection. For example:

```
var colorSamplerRef = app.activeDocument.colorSamplers[0];
var currentColor = colorSamplerRef.color;
```

Note: For additional information about color samplers, see Adobe Photoshop CS6 help on the Color SamplerTool.

Properties

Property	Value type	What it is
color	SolidColor	Read-only. The color of the color sampler.
position	array of UnitValue	Read-only. The position of the color sampler in the document. The array (x,y) represents the horizontal and vertical location of the count item.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `ColorSampler` object.

Methods

Method	Parameter type	Returns	What it does
move (position)	array of UnitValue		Moves the color sampler to a new location in the document. The `position` parameter (x,y) represents the new horizontal and vertical locations of the moved color sampler.
remove ()			Deletes the `ColorSampler` object.

ColorSamplers

The collection of ColorSampler objects in a document. Access through the Document.colorSamplers collection property. For example:

```
app.activeDocument.colorSamplers.removeAll()
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the ColorSamplers collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced ColorSamplers object.

Methods

Method	Parameter type	Returns	What it does
add (position)	array of UnitValue	ColorSampler	Creates a new color sampler object and adds it to this collection. The position parameter (x,y) represents the new horizontal and vertical locations of the moved color sampler.
removeAll ()			Removes all ColorSampler objects from the ColorSamplers collection.

ContactSheetOptions

Options for creating a contact sheet with the Application.makeContactSheet() method.

Properties

Property	Value type	What it is
acrossFirst	boolean	Read-write. True to place the images horizontally (left to right, then top to bottom) first (default: `true`).
bestFit	boolean	Read-write. True to rotate images for the best fit (default: `false`).
caption	boolean	Read-write. True to use the filename as a caption for the image (default: `true`).
columnCount	number[1..100]	Read-write. The number of columns to include (default: `5`).
flatten	boolean	Read-write. True to flatten all layers in the final document (default: `true`).
font	GalleryFontType	Read-write. The font used for the caption (default: `GalleryFontType.ARIAL`).
fontSize	number	Read-write. The font size to use for the caption (default: `12`).
height	number [0..29000]	Read-write. The height (in pixels) of the resulting document (default: `720`).
horizontal	number	Read-write. The horizontal spacing (in pixels) between images (default: `1`).
mode	NewDocumentMode	Read-write. The document color mode (default: `NewDocumentMode.RGB`).
resolution	number [35..1200]	Read-write. The resolution of the document in pixels per inch (default: `72.0`).
rowCount	number [1..100]	Read-write. The number of rows to use (default: `6`).
typename	string	Read-only. The class name of the referenced `contactSheetOptions` object.
useAutoSpacing	boolean	Read-write. True to auto space the images (default: `true`).
vertical	number [0..29000]	Read-write. The vertical spacing (in pixels) between images (default: `1`). Valid only when `useAutoSpacing = false`.
width	number [100..29000]	Read-write. The width (in pixels) of the resulting document (default: `576`).

CountItem

A counted item in a document. Access through the Document.countItems collection. See the Document.autoCount() method.

Note: This feature is available in the Extended Version only.

For additional information about count items, see Adobe Photoshop CS6 help on the Count Tool.

Properties

Property	Value type	What it is
position	array of UnitValue	Read-only. The position of the count item in the document.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `CountItem` object.

Methods

Method	Parameter type	Returns	What it does
remove ()			Deletes the `CountItem` object.

CountItems

The collection of CountItem objects in the document.

Access through the Document.countItems collection property. For example:

```
app.activeDocument.countItems.removeAll()
```

Note: This feature is available in the Extended Version only.

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `CountItems` collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `CountItems` object.

Methods

Method	Parameter type	Returns	What it does
add (position)	array of UnitValue	CountItem	Creates a new count item object and adds it to this collection. Parameter `position` (x,y) represents the horizontal and vertical positions, respectively, of the `CountItem` object.
getByName (name)	string	CountItem	Get the first element in the `CountItems` collection with the provided name.
removeAll ()			Removes all `CountItem` objects from the `CountItems` collection.

DCS1_SaveOptions

Options for saving a CMYK document in DCS1 format using the Document.saveAs() method.

Properties

Property	Value type	What it is
dCS	DCSType	Read-write. (default: `DCSType.COLORCOMPOSITE`).
embedColorProfile	boolean	Read-write. True to embed the color profile in the document
encoding	SaveEncoding	Read-write. The type of encoding to use for document (default: `SaveEncoding.BINARY`).
halftoneScreen	boolean	Read-write. True to include halftone screen (default: `false`).
interpolation	boolean	Read-write. True to use image interpolation (default: `false`)
preview	Preview	Read-write. The type of preview (default: `Preview.MACOSEIGHTBIT`).
transferFunction	boolean	Read-write. True to include the Transfer functions to compensate for dot gain between the image and film (default: `false`).
typename	string	Read-only. The class name of the referenced `DCS1_SaveOptions` object.
vectorData	boolean	Read-write. True to include vector data. Valid only if the document includes vector data (unrasterized text).

DCS2_SaveOptions

Options for saving a CMYK document in DCS2 format using the Document.saveAs() method.

Properties

Property	Value type	What it is
dCS	DCSType	Read-write. The type of composite file to create (default: `DCSType.NOCOMPOSITE`).
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
encoding	SaveEncoding	Read-write. The type of encoding to use (default: `SaveEncoding.BINARY`).
halftoneScreen	boolean	Read-write. True to include the halftone screen (default: `false`).
interpolation	boolean	Read-write. True to use image interpolation (default: `false`).
multiFileDCS	boolean	Read-write. True to save color channels as multiple files or a single file (default: `false`).
preview	Preview	Read-write. The preview type (default: `Preview.MACOSEIGHTBIT`).
spotColors	boolean	Read-write. True to save spot colors.
transferFunction	boolean	Read-write. True to include the Transfer functions to compensate for dot gain between the image and film (default: `false`).
typename	string	Read-only. The class name of the referenced `DCS2_SaveOptions` object.
vectorData	boolean	Read-write. True to include vector data. Valid only if the document includes vector data (unrasterized text).

DICOMOpenOptions

Options for opening a document in DICOM format using the Application.open() method.

Note: This feature is available in the Extended Version only.

Properties

Property	Value type	What it is
anonymize	boolean	Read-write. True to make the patient information anonymous.
columns	number	Read-write. Number of columns in n-up configuration.
reverse	boolean	Read-write. True to reverse (invert) the image.
rows	number	Read-write. The number of rows in n-up configuration.
showOverlays	boolean	Read-write. True to show overlays.
typename	string	Read-only. The class name of the referenced DICOMOpenOptions object.
windowLevel	number	Read-write. The contrast of the image in Houndsfield units.
windowWidth	number	Read-write. The brightness of the image in Houndsfield units.

Document

The active containment object for layers and all other objects in the script; the basic canvas for the file.

- Access the object for the currently active document through Application.activeDocument.
- You can access other documents, or iterate through all open documents using the list in the Application.documents collection. You can access individual documents in the list by index, or use Documents.getByName() to retrieve them by name.
- Create documents programmatically using the Documents.add() method.

See Document sample script and the Documents collection object for examples.

Note: In Adobe Photoshop CS6, a document can also be referred to as an image or a canvas.

- The term *image* refers to the entire document and its contents. You can trim or crop an image. You resize an image using the `resizeImage()` method.
- The term *canvas* refers to the space in which the document sits on the screen. You can rotate or flip the canvas. You resize the canvas using the `resizeCanvas()` method.

Properties

Property	Value type	What it is
activeChannels	array of Channel	Read-write. The selected channels.
activeHistoryBrushSource	Guide	Read-write. The history state to use with the history brush.
activeHistoryState	Guide	Read-write. The selected `HistoryState` object.
activeLayer	ArtLayer or LayerSet	Read-write. The selected layer.
artLayers	ArtLayers	Read-only. The art layers collection.
backgroundLayer	ArtLayer	Read-only. The background layer of the document.
bitsPerChannel	BitsPerChannelType	Read-write. The number of bits per channel.
channels	Channels	Read-only. The channels collection.
colorProfileName	string	Read-write. The name of the color profile. Valid only when colorProfileType = `ColorProfile.CUSTOM` or `WORKING`.
colorProfileType	ColorProfileType	Read-write. Whether the document uses the working color profile, a custom profile, or no profile.
colorSamplers	ColorSamplers	Read-only. The current color samplers associated with this document.
componentChannels	array of Channel	Read-only. The color channels that make up the document; for instance, the Red, Green, and Blue channels for an RGB document.

Property	Value type	What it is (Continued)
`countItems`	CountItems	Read-only. The current count items. **Note:** For additional information about count items, see Adobe Photoshop CS6 help on the Count Tool.
`fullName`	File	Read-only. The full path name of the document.
`guides`	Guides	Read-only. The guides collection.
`height`	UnitValue	Read-only. The height of the document (unit value).
`histogram`	array of number	Read-only. A histogram showing the number of pixels at each color intensity level for the composite channel. The array c ontains 256 members. Valid only when mode = `DocumentMode.RGB`, `CMYK`; or `INDEXEDCOLOR`.
`historyStates`	HistoryStates	Read-only. The history states collection.
`info`	DocumentInfo	Read-only. Metadata about the document.
`layerComps`	LayerComps	Read-only. The layer compositions collection.
`layers`	Layers	Read-only. The layers collection.
`layerSets`	LayerSets	Read-only. The layer set collection.
`managed`	boolean	Read-only. True if the document a is workgroup document.
`measurementScale`	MeasurementScale	Read-only. The measurement scale for the document. **Note:** The measurement scale feature is available in the Extended version only.
`mode`	DocumentMode	Read-only. The color profile.
`name`	string	Read-only. The document's name.
`parent`	Application	Read-only. The application object that contains this document.
`path`	File	Read-only. The path to the document.
`pathItems`	PathItems	Read-only. The path items collection.
`pixelAspectRatio`	number [0.100..10.000]	Read-write. The (custom) pixel aspect ratio to use.
`printSettings`	DocumentPrintSettings	Read-only. The print settings for the document.

Property	Value type	What it is (Continued)
quickMaskMode	boolean	Read-write. True if the document is in Quick Mask mode.
resolution	number	Read-only. The document's resolution (in pixels per inch).
saved	boolean	Read-only. True if the document has been saved since the last change.
selection	Selection	Read-only. The selected area of the document.
typename	string	Read-only. The class name of the `Document` object.
width	UnitValue	Read-only. The width of the document (unit value).
xmpMetadata	xmpMetadata	Read-only. XMP metadata for the document. Camera RAW settings for the image are stored here for example.

Methods

Method	Parameter type	Returns	What it does
autoCount (channel, threshold)	Channel number		Counts the number of objects in a document. Available in the Extended Version only. Creates a CountItem object for each object counted. For additional information about how to set up objects to count, see the Count Tool in the Adobe Photoshop CS6 Help
changeMode (destinationMode [, options])	ChangeMode BitmapConversionOptions or IndexedConversionOptions		Changes the color profile of the document.
close ([saving])	SaveOptions		Closes the document. If any changes have been made, the script presents an alert with three options: save, do not save, prompt to save. The optional parameter specifies a selection in the alert box (default: `SaveOptionsType.PROMPTTOSAVECHANGES`).
convertProfile (destinationProfile, intent [, blackPointCompensation] [, dither])	string Intent boolean boolean		Changes the color profile. The `destinationProfile` parameter must be either a string that names the color mode or `Working RGB`, `Working CMYK`, `Working Gray`, `Lab Color` (meaning one of the working color spaces or Lab color).
crop (bounds [, angle] [, width] [, height])	array of 4 UnitValue number UnitValue UnitValue		Crops the document. The `bounds` parameter is an array of four coordinates for the region remaining after cropping, [left, top, right, bottom].
duplicate ([name] [, mergeLayersOnly])	string boolean	Document	Creates a duplicate of the `document` object. The optional parameter `name` provides the name for the duplicated document. The optional parameter `mergeLayersOnly` indicates whether to only duplicate merged layers.

Method	Parameter type	Returns	What it does (Continued)
exportDocument (exportIn [, exportAs] [, options])	File ExportType ExportOptionsIllustrator —or— ExportOptionsSaveForWeb		Exports the paths in the document to an Illustrator file, or exports the document to a file with Web or device viewing optimizations. This is equivalent to choosing **File > Export > Paths To Illustrator**, or **File > Save For Web and Devices**.
flatten ()			Flattens all layers in the document.
flipCanvas (direction)	Direction		Flips the image within the canvas in the specified direction.
importAnnotations (file)	File		Imports annotations into the document.
mergeVisibleLayers ()			Flattens all visible layers in the document.
paste ([intoSelection])	boolean	ArtLayer	Pastes the contents of the clipboard into the document. If the optional argument is set to `true` and a selection is active, the contents are pasted into the selection.
print ([sourceSpace] [, printSpace] [, intent] [blackPointCompensation])	SourceSpaceType string Intent boolean		Prints the document. `printSpace` specifies the color space for the printer. Valid values are `nothing` (that is, the same as the source); or `Working RGB`, `Working CMYK`, `Working Gray`, `Lab Color` (meaning one of the working color spaces or Lab color); or a string specifying a specific colorspace (default is same as source).
printOneCopy ()			Print one copy of the document.
rasterizeAllLayers ()			Rasterizes all layers.
recordMeasurements ([source] [, dataPoints])	MeasurementSource array of string		Record measurements of document.
resizeCanvas ([width] [, height] [, anchor])	UnitValue UnitValue AnchorPosition		Changes the size of the canvas to display more or less of the image but does not change the image size. See resizeImage.
resizeImage ([width] [, height] [, resolution] [, resampleMethod])	UnitValue UnitValue number ResampleMethod		Changes the size of the image.

Method	Parameter type	Returns	What it does (Continued)
revealAll ()			Expands the document to show clipped sections.
rotateCanvas (angle)	number		Rotates the canvas (including the image) in clockwise direction.
save ()			Saves the document.
saveAs (saveIn [, options] [, asCopy] [, extensionType])	File object (see description) boolean Extension		Saves the document in a specific format. Specify the save options appropriate to the format by passing one of these objects: BMPSaveOptions DCS1_SaveOptions DCS2_SaveOptions EPSSaveOptions GIFSaveOptions JPEGSaveOptions PDFSaveOptions PhotoshopSaveOptions PICTFileSaveOptions PICTResourceSaveOptions PixarSaveOptions PNGSaveOptions RawSaveOptions SGIRGBSaveOptions TargaSaveOptions TiffSaveOptions
splitChannels ()		array of Document	Splits the document channels into separate images.
suspendHistory (historyString javaScriptString)	string string		Provides a single entry in history states for the entire script provided by `javaScriptString`. Allows a single undo for all actions taken in the script. The `historyString` parameter provides the string to use for the history state. The `javaScriptString` parameter provides a string of JavaScript code to excute while history is suspended.
trap (width)	number		Applies trapping to a CMYK document. Valid only when *docRef*.mode = DocumentMode.CMYK.
trim ([type] [, top] [, left] [, bottom] [, right])	TrimType boolean boolean boolean boolean		Trims the transparent area around the image on the specified sides of the canvas. Default is `true` for all Boolean parameters.

Document sample script

The following script creates a document that contains two images (a sunflower and a duck) obtained from the Adobe Photoshop CS6 Samples folder and employs the following steps:

- Determines which image is larger.
- Resizes the smaller image to match the larger image.
- Creates a merged document twice as high as either image in order to hold both images.
- Selects part of the document and pastes the sunflower into the selection.
- Inverts the selection and pastes the duck into the lower part of the document.
- Positions the sunflower over the duck.

Document.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// first close all the open documents
while (app.documents.length) {
   app.activeDocument.close()
   }

// Open the sunflower and duck files from the samples folder
var flowerDoc = open(File(app.path + "/Samples/Sunflower.psd"))
var duckDoc = open(File(app.path + "/Samples/Ducky.tif"))

// Find out which document is larger
// Resize the smaller document the to the larger document's size
// The resize requires the document be the active/front document
if ((flowerDoc.width.value * flowerDoc.height.value) >
      (duckDoc.width.value * duckDoc.height.value)) {
   app.activeDocument = duckDoc
   duckDoc.resize(flowerDoc.width, flowerDoc.height)
   }
else {
   app.activeDocument = flowerDoc
   flowerDoc.resizeImage(duckDoc.width, duckDoc.height)
   }

// Create a new document twice as high as two files
var mergedDoc = app.documents.add(duckDoc.width, duckDoc.height * 2,
duckDoc.resolution, "FlowerOverDuck")

// Copy the flower to the top; make it the active document so we can manipulate it
app.activeDocument = flowerDoc
flowerDoc.activeLayer.copy()

//Paste the flower to the merged document, making the merged document active
app.activeDocument = mergedDoc

// Select a square area at the top of the new document
```

```
var selRegion = Array(Array(0, 0),
                   Array(mergedDoc.width.value, 0),
                   Array(mergedDoc.width.value, mergedDoc.height.value / 2),
                   Array(0, mergedDoc.height.value / 2),
                   Array(0, 0))
// Create the selection
mergedDoc.selection.select(selRegion)

//Paste in the flower
mergedDoc.paste(TRUE)

// do the same thing for the duck
app.activeDocument = duckDoc
duckDoc.activeLayer.copy()

app.activeDocument = mergedDoc
mergedDoc.selection.select(selRegion)

// Inverting the selection so the bottom of the document is now selected
mergedDoc.selection.invert()

// Paste the duck
mergedDoc.paste(TRUE)

// get rid of our originals without modifying them
duckDoc.close(SaveOptions.DONOTSAVECHANGES)
flowerDoc.close(SaveOptions.DONOTSAVECHANGES)

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs
```

DocumentPrintSettings

The print settings for a document.

Properties

Property	Value type	What it is
backgroundColor	SolidColor	Read-write. Background color of page.
bleedWidth	UnitValue	Read-write. Bleed width
caption	boolean	Read-write. Print the caption found in FileInfo.
centerCropMarks	boolean	Read-write. Print center crop marks.
colorBars	boolean	Read-write. Print color calibration bars.
copies	number	Read-write. Number of copies to print.
cornerCropMarks	boolean	Read-write. Print corner crop marks.
colorHandling	PrintColorHandling	Read-only. Color handling.
activePrinter	string	Read-write. The currently active printer.
flip	boolean	Read-write. Flip the image horizontally.
hardProof	boolean	Read-write. Print a hard proof.
interpolate	boolean	Read-write.
labels	boolean	Read-write. Prints the document title.
mapBlack	boolean	Read-write. Map blacks.
negative	boolean	Read-write. Invert the image colors.
renderIntent	Intent	Read-write. Color conversion intent when print space is different from the source space.
posX	UnitValue	Read-only. The x position of the image on page.
posY	UnitValue	Read-only. The y position of the image on page.
printBorder	UnitValue	Read-write. The width of the print border.
printerName	string	Read-write. Name of the printer.
printSpace	string	Read-write. color space for printer. Can be nothing (meaning same as source); 'Working RGB', 'Working CMYK', 'Working Gray', 'Lab Color' (meaning one of the working spaces or Lab color); or a string specifying a specific colorspace (default is same as source)
registrationMarks	boolean	Read-write. Print registration marks.

Property	Value type	What it is (Continued)
scale	number	Read-only. Scale of image on page.
vectorData	boolean	Read-write. Include vector data.

Methods

Method	Parameter type	Returns	What it does
setPagePosition (docPosition, posX, posY, scale)	DocPositionStyle UnitValue UnitValue number		Set the position of the image on the page.

DocumentInfo

Metadata about a `document` object.

Access through the Document.info property. For example, the following sets the `author`, `caption`, and `copyrighted` properties:

```
var docRef = open(fileList[i])
// set the file info
docRef.info.author = "Mr. Adobe programmer"
docRef.info.caption = "Adobe Photo shoot"
docRef.info.copyrighted = CopyrightedType.COPYRIGHTEDWORK
```

These values can be set interactively by choosing **File > File Info**.

Properties

Property	Value type	What it is
author	string	Read-write.
authorPosition	string	Read-write.
caption	string	Read-write.
captionWriter	string	Read-write.
category	string	Read-write.
city	string	Read-write.
copyrighted	CopyrightedType	Read-write. The copyrighted status.
copyrightNotice	string	Read-write.
country	string	Read-write.
creationDate	string	Read-write.
credit	string	Read-write.
exif	array of array [*tag data*]	Read-only. Camera data that includes camera settings used when the image was taken. Each array member is a tag pair, an array of [*tag*, *tag_data*]; for example, `[ "camera" "Cannon"]`.
headline	string	Read-write.
instructions	string	Read-write.
jobName	string	Read-write.
keywords	array of string	Read-write. A list of keywords that can identify the document or its contents.
ownerUrl	string	Read-write.

Property	Value type	What it is (Continued)
`parent`	`Document`	Read-only. The `info` object's container.
`provinceState`	`string`	Read-write.
`source`	`string`	Read-write.
`supplementalCategories`	`array of string`	Read-write.
`title`	`string`	Read-write.
`transmissionReference`	`string`	Read-write.
`typename`	`string`	Read-only. The class name of the referenced `info` object.
`urgency`	`Urgency`	Read-write.

DocumentInfo sample Script

The following script sets document info (metadata) for all of the files in a specified folder and then saves the modified files as low-quality JPEG images in a new folder without changing the originals.

- Ask the user to specify the folder that contains the original files and the output folder for the JPEG images, and then check that the folders exist.
- Open each file and use the `documentInfo` object properties to tag it with the following metadata:
  - author: Adobe programmer
  - caption: Adobe Photo shoot
  - captionWriter: Adobe programmer
  - city: San Jose
  - copyrightNotice: Copyright (c) Adobe programmer Photography
  - copyrighted status: Copyrighted Work
  - country: USA
  - state: CA
- Save the new documents in JPEG format with a low quality setting.

DocumentInfo.jsx

```
// Save the current preferences
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.displayDialogs = DialogModes.NO

// ask the user for the input and output folders
var inputFolder = Folder.selectDialog("Select a folder to tag")
var outputFolder = Folder.selectDialog("Select a folder for the output files")

// see if we got something interesting from the dialog
if (inputFolder != null && outputFolder != null) {
   // get all the files found in this folder
   var fileList = inputFolder.getFiles()
```

```
    // save the outputs in JPEG
    var jpegOptions = new JPEGSaveOptions()
    // set the jpeg quality really low so the files are small
    jpegOptions.quality = 1
    // open each one in turn
    for (var i = 0; i < fileList.length; i++) {
        // The fileList includes both folders and files so open only files
        if (fileList[i] instanceof File && fileList[i].hidden == false) {
            // get a reference to the new document
            var docRef = open(fileList[i])

            // tag all of the documents with photo shoot information
            docRef.info.author = "Adobe programmer"
            docRef.info.caption = "Adobe Photo shoot"
            docRef.info.captionWriter = "Adobe programmer"
            docRef.info.city = "San Jose"
            docRef.info.copyrightNotice = "Copyright (c) Adobe programmer
                Photography"
            docRef.info.copyrighted = CopyrightedType.COPYRIGHTEDWORK
            docRef.info.country = "USA"
            docRef.info.provinceState = "CA"

            // change the date to a Adobe Photoshop CS6 date format
            // "YYYYMMDD"
            var theDate = new Date()
            // the year is from 1900 ????
            var theYear = (theDate.getYear() + 1900).toString()
            // convert the month from 0..12 to 00..12
            var theMonth = theDate.getMonth().toString()
            if (theDate.getMonth() < 10) {
                theMonth = "0" + theMonth
                }
            // convert the day from 0..31 to 00.31
            var theDay = theDate.getDate().toString()
            if (theDate.getDate() < 10) {
                theDay = "0" + theDay
                }
            // stick them all together
            docRef.info.creationDate = theYear + theMonth + theDay

            // flatten because we are saving to JPEG
            docRef.flatten()
            // go to 8 bit because we are saving to JPEG
            docRef.bitsPerChannel = BitsPerChannelType.EIGHT

            // save and close
            docRef.saveAs(new File(outputFolder + "/Output" + i + ".jpg"), jpegOptions)

            // don't modify the original
            docRef.close(SaveOptions.DONOTSAVECHANGES)
        }
    }
}

// Reset the application preferences
app.displayDialogs = startDisplayDialogs
```

Documents

The collection of open Document objects.

Access this list through the Application.documents collection property, which is available through the app global variable, or directly at the top level. For example, the following adds a new document to the collection:

```
app.documents.add(800, 500, 72, "myDocument", NewDocumentMode.RGB)
```

—or—

```
documents.add(800, 500, 72, "myDocument", NewDocumentMode.RGB)
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the documents collection.
parent	Application	Read-only. The containing application.
typename	string	Read-only. The class name of the referenced documents object.

Methods

Method	Parameter type	Returns	What it does
add ([width] [, height] [, resolution] [, name] [, mode] [, initialFill] [,pixelAspectRatio] [, bitsPerChannel] [,colorProfileName])	 UnitValue UnitValue number string NewDocumentMode DocumentFill number [0.1..10.00] BitsPerChannelType string	Document	Creates a new document object and adds it to this collection. pixelAspectRatio: Default is 1.0, a square aspect ratio. bitsPerChannelType: Default is BitsPerChannelType.EIGHT.
getByName (name)	string	Document	Gets the first element in the documents collection with the provided name

EPSOpenOptions

Options for opening a document in EPS format using the Application.open() method.

Properties

Property	Value type	What it is
antiAlias	boolean	Read-write. True to use antialias.
constrainProportions	boolean	Read-write. True to constrain the proportions of the image.
height	UnitValue	Read-write. The height of the image (unit value).
mode	OpenDocumentMode	Read-write. The color profile to use as the document mode.
resolution	number	Read-write. The resolution of the document in pixels per inch.
typename	string	Read-only. The class name of the referenced `EPSOpenOptions` object.
width	UnitValue	Read-write. The width of the image (unit value).

EPSSaveOptions

Options for saving a document in EPS format using the Document.saveAs() method.

Properties

Property	Value type	What it is
embedColorProfile	boolean	Read-write. True to embed the color profile in this document.
encoding	SaveEncoding	Read-write. The type of encoding to use (default: `SaveEncoding.BINARY`).
halftoneScreen	boolean	Read-write. True to include the halftone screen (default: `false`).
interpolation	boolean	Read-write. True to use image interpolation (default: `false`).
preview	Preview	Read-write. The preview type.
psColorManagement	boolean	Read-write. True to use Postscript color management (default: `false`).
transferFunction	boolean	Read-write. True to include the Transfer functions to compensate for dot gain between the image and film (default: `false`).
transparentWhites	boolean	Read-write. True to display white areas as transparent. Valid only when *document*.mode = `DocumentMode.BITMAP`. See also changeMode().
typename	string	Read-only. The class name of the referenced `EPSSaveOptions` object.
vectorData	boolean	Read-write. True to include vector data. Valid only if the document includes vector data (text).

ExportOptionsIllustrator

Options for exporting PathItem objects to an Adobe Illustrator® file using using the Document.exportDocument() method. These options are the options that you can provide when you choose **File > Export > Paths To Illustrator**.

Properties

Property	Value type	What it is
path	IllustratorPathType	Read-write. The type of path to export (default: IllustratorPathType.DOCUMENTBOUNDS).
pathName	string	Read-write. The name of the path to export. Valid only when path = IllustratorPathType.NAMEDPATH.
typename	string	Read-only. The class name of the referenced exportOptionsIllustrator object.

ExportOptionsSaveForWeb

Options for optimizing a document for the web or devices using the Document.exportDocument() method. These are the options that you can provide when you choose **File > Save For Web and Devices**.

Properties

Property	Value type	What it is
blur	number	Read-write. Applies blur to the image to reduce artifacts (default: `0.0`).
colorReduction	ColorReductionType	Read-write. The color reduction algorithm (default: `ColorReductionType.SELECTIVE`).
colors	number	Read-write. The number of colors in the palette (default: `256`).
dither	Dither	Read-write. The type of dither (default: `Dither.DIFFUSION`).
ditherAmount	number	Read-write. The amount of dither (default: 100). Valid only when dither `= Dither.DIFFUSION`.
format	SaveDocumentType	Read-write. The file format to use (default: `SaveDocumentType.COMPUSERVEGIF`). **Note:** For this property, only `COMPUSERVEGIF`, `JPEG`, `PNG-8`, `PNG-24`, and `BMP` are supported.
includeProfile	boolean	Read-write. True to include the document's embedded color profile (default: `false`).
interlaced	boolean	Read-write. True to download in multiple passes; progressive (default: `false`).
lossy	number	Read-write. The amount of lossiness allowed (default: `0`).
matteColor	RGBColor	Read-write. The colors to blend transparent pixels against.
optimized	boolean	Read-write. True to create smaller but less compatible files (default: `true`). Valid only when format `= SaveDocumentType.JPEG`.
PNG8	boolean	Read-write. Indicates the number of bits; `true` = 8, `false` = 24 (default: `true`). Valid only when format `= SaveDocumentType.PNG`.

Property	Value type	What it is (Continued)
quality	number [0..100]	Read-write. The quality of the produced image as a percentage; default: 60.
transparency	boolean	Read-write. Indication of transparent areas of the image should be included in the saved image(default: true).
transparencyAmount	number	Read-write. The amont of transparency dither (default: 100). Valid only if transparency = true.
transparencyDither	Dither	Read-write. The transparency dither algorithm (default: transparencyDither = Dither.NONE).
typename	string	Read-only. The class name of the referenced ExportOptionsSaveForWeb object.
webSnap	number	Read-write. The tolerance amount within which to snap close colors to web palette colors (default: 0).

File

Folder

ExtendScript defines the JavaScript classes `File` and `Folder` to encapsulate file-system references in a platform-independent manner; see 'JavaScript support in Adobe Photoshop CS6' on page 32. For references details of these classes, see the *JavaScript Tools Guide*.

GalleryBannerOptions

Options for the bannerOptions property of the GalleryOptions object.

Tip: You can preserve default values for many of these properties by setting the GalleryOptions property preserveAllMetadata to true; this is the same as choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
contactInfo	string	Read-write. The web photo gallery contact info.
date	string	Read-write. The web photo gallery date (default: current date).
font	GalleryFontType	Read-write. The font setting for the banner text (default: GalleryFontType.ARIAL).
fontSize	number [1..7]	Read-write. The font size for the banner text (default: 3).
photographer	string	Read-write. The web photo gallery photographer.
siteName	string	Read-write. The web photo gallery site name (default: Adobe Web Photo Gallery).
typename	string	Read-only. The class name of the referenced galleryBannerOptions object.

GalleryCustomColorOptions

Options for the customColorOptions property of the GalleryOptions object.

Tip: You can preserve default values for many of these properties by setting the GalleryOptions property preserveAllMetadata to true; this is the same as choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
activeLinkColor	RGBColor	Read-write. The color to use to indicate an active link.
backgroundColor	RGBColor	Read-write. The background color.
bannerColor	RGBColor	Read-write. The banner color.
linkColor	RGBColor	Read-write. The color to use to indicate a link.
textColor	RGBColor	Read-write. The text color.
typename	string	Read-only. The class name of the referenced galleryCustomColorOptions object.
visitedLinkColor	RGBColor	Read-write. The color to use to indicate a visited link.

GalleryImagesOptions

Options for the imagesOptions property of the GalleryOptions object.

Tip: You can preserve default values for many of these properties by setting the GalleryOptions property preserveAllMetadata to true; this is the same as choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
border	`number [0..99]`	Read-write. The size (in pixels) of the border that separates images (default: `0`).
caption	`boolean`	Read-write. True to generate image captions (default: `false`).
dimension	`number`	Read-write. The resized image dimensions in pixels (default: `350`). Valid only when resizeImages `= true`.
font	`GalleryFontType`	Read-write. The font to use for image captions (default: `GalleryFontType.ARIAL`).
fontSize	`number [1..7]`	Read-write. The font size for image captions (default: `3`). Valid only when caption `= true`.
imageQuality	`number [0..12]`	Read-write. The quality setting for a JPEG image (default: `5`).
includeCopyright	`boolean`	Read-write. True to include copyright information in captions (default: `false`). Valid only when caption `= true`.
includeCredits	`boolean`	Read-write. True to include the credits in image captions (default: `false`). Valid only when caption `= true`.
includeFilename	`boolean`	Read-write. True to include the file name in image captions (default: `true`). Valid only when caption `= true`.
includeTitle	`boolean`	Read-write. True to include the title in image captions (default: `false`). Valid only when caption `= true`.
numericLinks	`boolean`	Read-write. True to add numeric links (default: `true`).

Property	Value type	What it is (Continued)
`resizeConstraint`	`GalleryConstrainType`	Read-write. The image dimensions to constrain in the gallery image (default: `GalleryConstrainType.CONSTRAINBOTH`). Valid only when resizeImages `= true`.
`resizeImages`	`boolean`	Read-write. True to automatically resize images for placement on the gallery pages (default: `true`).
`typename`	`string`	Read-only. The class name of the referenced `galleryImagesOptions` object.

GalleryOptions

Options for a Web photo gallery, created with Application.makePhotoGallery().

Tip: You can preserve default values for many of these properties by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
addSizeAttributes	boolean	Read-write. True to add width and height attributes for images (default: `true`).
bannerOptions	GalleryBannerOptions	Read-write. The options related to banner settings.
customColorOptions	GalleryCustomColorOptions	Read-write. The options related to custom color settings.
emailAddress	string	Read-write. The email address to show on the web page.
imagesOptions	GalleryImagesOptions	Read-write. The options related to images settings.
includeSubFolders	boolean	Read-write. True to include all files found in sub folders of the input folder (default: `true`).
layoutStyle	string	Read-write. The style to use for laying out the web page (default: `Centered Frame 1 - Basic`).
preserveAllMetadata	boolean	Read-write. True to save metadata (default: `false`).
securityOptions	GallerySecurityOptions	Read-write. The options related to security settings.
thumbnailOptions	GalleryThumbnailOptions	Read-write. The options related to thumbnail image settings.
typename	string	Read-only. The class name of the referenced `galleryOptions` object.
useShortExtension	boolean	Read-write. True to use the short web page extension `.htm`. If false, use the web page extension `.html` (default: `true`).
useUTF8Encoding	boolean	Read-write. True to use UTF-8 encoding for the web page (default: `false`).

GallerySecurityOptions

Options for the securityOptions property of the GalleryOptions object.

Tip: You can preserve default values for many of these properties by setting the GalleryOptions property preserveAllMetadata to true; this is the same as choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
`content`	`GallerySecurityType`	Read-write. The web photo gallery security content (default: `GallerySecurityType.NONE`).
`font`	`GalleryFontType`	Read-write. The web photo gallery security font (default: `GalleryFontType.ARIAL`).
`fontSize`	`number [1..72]`	Read-write. The web photo gallery security font size (default: `3`).
`opacity`	`number`	Read-write. The web page security opacity as a percent (default: `100`).
`text`	`string`	Read-write. The web photo gallery security custom text.
`textColor`	`GallerySecurityTextColorType`	Read-write. The web page security text color.
`textPosition`	`GallerySecurityTextPositionType`	Read-write. The web photo gallery security text position (default: `GallerySecurityTextPositionType.CENTERED`).
`textRotate`	`GallerySecurityTextRotateType`	Read-write. The web photo gallery security text orientation to use (default: `GallerySecurityTextRotateType.ZERO`).
`typename`	`string`	Read-only. The class name of the referenced `gallerySecurityOptions` object.

GalleryThumbnailOptions

Options for the thumbnailOptions property of the GalleryOptions object.

Tip: You can preserve default values for many of these properties by setting the GalleryOptions property preserveAllMetadata to true; this is the same as choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** in the Options area of the Web Photo Gallery dialog.

Properties

Property	Value type	What it is
border	number [0..99]	Read-write. The amount of border pixels you want around your thumbnail images (default: 0).
caption	boolean	Read-write. True if there is a caption (default: false).
columnCount	number	Read-write. The number of columns on the page (default: 5).
dimension	number	Read-write. The web photo gallery thumbnail dimension in pixels (default: 75).
font	GalleryFontType	Read-write. The web photo gallery font (default: GalleryFontType.ARIAL).
fontSize	number [1..7]	Read-write. The font size for thumbnail images text (default: 3).
includeCopyright	boolean	Read-write. True to include copyright information for thumbnails (default: false).
includeCredits	boolean	Read-write. True to include credits for thumbnails (default: false).
includeFilename	boolean	Read-write. True to include file names for thumbnails (default: false).
includeTitle	boolean	Read-write. True to include titles for thumbnails (default: false).
rowCount	number	Read-write. The number of rows on the page (default: 3).
size	GalleryThumbSizeType	Read-write. The thumbnail image size (default: GalleryThumbSizeType.MEDIUM).
typename	string	Read-only. The class name of the referenced GalleryThumbnailOptions object.

GIFSaveOptions

Options for saving a document in GIF format using the Document.saveAs() method.

Properties

Property	Value type	What it is
colors	number	Read-write. The number of palette colors. Valid only when palette = Palette.LOCALADAPTIVE, LOCALPERCEPTUAL, LOCALSELECTIVE, MACOSPALETTE, UNIFORM, WEBPALETTE; or WINDOWSPALETTE .
dither	Dither	Read-write. The dither type.
ditherAmount	number [1..100]	Read-write. The amount of dither (default: 75). Valid only when dither = Dither.DIFFUSION.
forced	ForcedColors	Read-write. The type of colors to force into the color palette.
interlaced	boolean	Read-write. True if rows should be interlaced (default: false).
matte	MatteType	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: MatteType.WHITE). When transparency = false, the matte color is applied to transparent areas.
palette	PaletteType	Read-write. The type of palette to use (default: Palette.LOCALSELECTIVE).
preserveExactColors	boolean	Read-write. True to protect colors in the image that contain entries in the color table from being dithered. Valid only when dither = Dither.DIFFUSION.
transparency	boolean	Read-write. True to preserve transparent areas of the image during conversion to GIF format.
typename	string	Read-only. The class name of the referenced GIFSaveOptions object.

GrayColor

Defines a gray color, used in the SolidColor object.

See also CMYKColor, HSBColor, LabColor, NoColor, RGBColor

Properties

Property	Value type	What it is
gray	number [0.0..100.0]	Read-write. The gray value (default: 0.0).
typename	string	Read-only. The class name of the referenced grayColor object.

Guide

Properties

Property	Value type	What it is
`direction`	Direction	Read-write. Indicates whether the guide is vertical or horizontal.
`coordinate`	UnitValue	Read-write. Location of the guide from origin of image.

Guides

The collection of open Guide objects.

Access this list through the Document.guides collection property, which is available through the `activeDocument` object. For example, the following adds a new guide to the collection:

```
activeDocument.guides.add (Direction.HORIZONTAL,UnitValue(20,20))
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `guides` collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `guides` object.

Methods

Method	Parameter type	Returns	What it does
add (direction , coordinate)	Direction UnitValue	Guide	Creates a new guide object and adds it to this collection.
getByName (name)	string	Guide	Gets the first element in the `guides` collection with the provided name

HistoryState

A version of the document stored automatically (and added to the HistoryStates collection), which preserves the document's state, each time the document is changed.

Access through Document.historyStates collection. You can access a state in the list by name. For example, this assigns a property value in the state object named "AddLayerMask":

```
var stateRef = app.activeDocument.historyState.getByName("AddLayerMask");
stateRef.snapshot = true;
```

Properties

Property	Value type	What it is
name	string	Read-only. The HistoryState object's name.
parent	Document	Read-only. The containing document.
snapshot	boolean	Read-only. True if the history state is a snapshot.
typename	string	Read-only. The class name of the referenced HistoryState object.

HistoryStates

The collection of Guide objects in the document.

Access through Document.historyStates collection property. For example, this accesses one of the states in the collection by index:

```
myState = app.activeDocument.historyStates[7];
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `HistoryStates` collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced `HistoryStates` object.

Methods

Method	Parameter type	Returns	What it does
getByName (name)	string	Guide	Get the first element in the `HistoryStates` collection with the provided name.

HSBColor

Defines an HSB color, used in the SolidColor object.

See also CMYKColor, GrayColor, LabColor, NoColor, RGBColor

Properties

Property	Value type	What it is
brightness	number[0.0..100.0]	Read-write. The brightness value.
hue	number [0.0..360.0]	Read-write. The hue value.
saturation	number [0.0..100.0]	Read-write. The saturation value.
typename	string	Read-only. The class name of the referenced HSBColor object.

IndexedConversionOptions

Options for converting an RGB image to an indexed color model using Document.changeMode().

Properties

Property	Value type	What it is
colors	number	Read-write. The number of palette colors. Valid only when palette = Palette.LOCALADAPTIVE, LOCALPERCEPTUAL, LOCALSELECTIVE, MACOSPALETTE, UNIFORM, WEBPALETTE, or WINDOWSPALETTE.
dither	Dither	Read-write. The dither type.
ditherAmount	number [1..100]	Read-write. The amount of dither. Valid only when dither = Dither.diffusion.
forced	ForcedColors	Read-write. The type of colors to force into the color palette.
matte	MatteType	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: MatteType.WHITE). When transparency = false, the matte color is applied to transparent areas.
palette	PaletteType	Read-write. The palette type (default: Palette.EXACT).
preserveExactColors	boolean	Read-write. True to protect colors in the image that contain entries in the color table from being dithered. Valid only when dither = Dither.DIFFUSION.
transparency	boolean	Read-write. True to preserve transparent areas of the image during conversion to GIF format.
typename	string	Read-only. The class name of the referenced IndexedConversionOptions object.

JPEGSaveOptions

Options for saving a document in JPEG format using the Document.saveAs() method.

Properties

Property	Value type	What it is
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
formatOptions	FormatOptions	Read-write.The download format to use (default: FormatOptions.STANDARDBASELINE).
matte	MatteType	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: MatteType.WHITE). When transparency is turned off for an image, the matte color is applied to transparent areas.
quality	number [0..12]	Read-write. The image quality setting to use; affects file size and compression (default: 3).
scans	number [3..5]	Read-write. The number of scans to make to incrementally display the image on the page (default: 3). Valid only for when formatOptions = FormatOptions.PROGRESSIVE.
typename	string	Read-only. The class name of the referenced JPEGSaveOptions object.

LabColor

Defines an LAB color, used in the SolidColor object.

See also CMYKColor, GrayColor, HSBColor, NoColor, RGBColor

Properties

Property	Value type	What it is
a	number [-128.0..127.0]	Read-write. The a-value.
b	number [-128.0..127.0]	Read-write. The b-value.
l	number [0.0..100.0]	Read-write. The L-value.
typename	string	Read-only. The class name of the referenced LabColor object.

LayerComp

A snapshot of a state of the layers in a document, which can be used to view different page layouts or compositions.

Access through Document.layerComps collection. You can access a layer comp by its name. For example, this sets the `comment` property value for a `LayerComp` object named `myLayerComp`:

```
var layercompRef = app.activeDocument.layerComps.getByName("myLayerComp");
layercompRef.comment = "View from shoreline";
```

Properties

Property	Value type	What it is
appearance	boolean	Read-write. True to use layer appearance (layer styles) settings.
comment	string	Read-write. A description of the layer comp.
name	string	Read-write. The name of the layer comp.
parent	Document	Read-write. The containing document.
position	boolean	Read-write. True to use layer position.
selected	boolean	Read-only. True if the layer comp is currently selected.
typename	string	Read-only. The class name of the referenced `layerComp` object.
visibility	boolean	Read-write. True to use layer visibility settings .

Methods

Method	Parameter type	Returns	What it does
apply ()			Applies the layer comp to the document.
recapture ()			Recaptures the current layer state(s) for this layer comp.
remove ()			Deletes the `layerComp` object.
resetfromComp ()			Resets the layer comp state to the document state.

LayerComps

The collection of LayerComp objects in the document.

Access through the Document.layerComps collection property. For example:

```
app.activeDocument.layerComps.add("myLayerComp", "View from Shoreline",
   true, true, true);
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the layerComps collection.
parent	Document	Read-only. The containing document.
typename	string	Read-only. The class name of the referenced layerComps object.

Methods

Method	Parameter type	Returns	What it does
add (name, comment, appearance, position, visibility)	 string string boolean boolean boolean	LayerComp	Creates a new layer composition object and adds it to this collection.
getByName (name)	 string	LayerComp	Gets the first element in the collection with the provided name.
removeAll ()			Removes all member objects from the layerComps collection.

Layers

The collection of layer objects, including ArtLayer and LayerSet objects, in the document. Access through Document.layers, or the LayerSet.layers collections properties.

For example, this uses the `length` property to count the number of `layer` objects in the active document, then displays the number on the screen:

```
var layerNum = app.activeDocument.layers.length
alert(layerNum)
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `layers` collection.
parent	Document or LayerSet	Read-only. The containing document or layer set.
typename	string	Read-only. The class name of the referenced `layers` object.

Methods

Method	Parameter type	Returns	What it does
getByName (name)	string	Layer	Gets the first element in the `layers` collection with the provided name.
removeAll ()			Removes all layers from the collection.

LayerSet

A group of layer objects, which can include ArtLayer objects and other (nested) `LayerSet` objects. A single command can manipulate all layers in the set.

Access top-level layers sets in a document through the Document.layerSets collection. You can access a layer set by name. For example, the following sets the `allLocked` value for "myLayerSet":

```
var layerSetRef = app.activeDocument.layerSets.getByName("myLayerSet");
layerSetRef.allLocked = true
```

Access nested layer sets through the LayerSet.layerSets collection in the parent set. For example:

```
app.activeDocument.layerSets[0].layerSets[0];
```

Properties

Property	Value type	What it is
allLocked	boolean	Read-write. True if the contents in the layers in this set are not editable.
artLayers	ArtLayers	Read-only. The art layers in this layer set.
blendMode	BlendMode	Read-write. The blend mode to use for the layer set.
bounds	array of UnitValue	Read-only. The bounding rectangle of the layer set.
enabledChannels	array of Channel	Read-write. The channels enabled for the layer set; must be a list of component channels. See Channel.kind.
layers	Layers	Read-only. The layers in this layer set.
layerSets	LayerSets	Read-only. Nested layer sets contained within this layer set.
linkedLayers	array of ArtLayer and/or LayerSet	Read-only. The layers linked to this `layerSet` object.
name	string	Read-write. The name of this layer set.
opacity	number [0.0..100.0]	Read-write. The master opacity of the set.
parent	Document or LayerSet	Read-only. The containing document or layer set.
typename	string	Read-only. The class name of the referenced `LayerSet` object.
visible	boolean	Read-write. True if the set is visible.

Methods

Method	Parameter type	Returns	What it does
duplicate ([relativeObject] [, insertionLocation])	ArtLayer or LayerSet ElementPlacement	LayerSet	Creates a duplicate of the object.
link (with)	ArtLayer or LayerSet		Links the layer set with another layer.
merge ()		ArtLayer	Merges the layerset; returns a reference to the art layer created by this method.
move (relativeObject, insertionLocation)	ArtLayer or LayerSet ElementPlacement		Moves the object.
remove ()			Deletes the object.
resize ([horizontal] [, vertical] [, anchor])	number number AnchorPosition		Resizes all layers in the layer set to to the specified dimensions (as a percentage of its current size) and places the layer set in the specified position.
rotate (angle [, anchor])	number AnchorPosition		Rotates all layers in the layer set around the specified anchor point (default: `AnchorPosition.MIDDLECENTER`)
translate ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the position relative to its current position.
unlink ()			Unlinks the layer set.

LayerSets

The collection of LayerSet objects in the document.

Access the top-level layer sets in a document through the Document.layerSets collection property. For example:

```
var layerSetRef = app.activeDocument.layerSets.add()
```

Access the nested layer sets through the LayerSet.layerSets collection property in the parent set. For example:

```
var layerSetRef = app.activeDocument.layerSets.getByName("myParentSet");
var childSet = layerSetRef.layerSets.getByName("myChildSet");
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `LayerSets` collection.
parent	Document or LayerSet	Read-only. The containing document or layer set.
typename	string	Read-only. The class name of the referenced `layerSets` object.

Methods

Method	Parameter type	Returns	What it does
add ()		LayerSet	Creates a new layer set object and adds it to the collection.
getByName (name)	string	LayerSet	Gets the first element in the collection with the provided name.
removeAll ()			Removes all member layer sets, and any layers or layer sets they contain, from the document.

LayerSets sample script

The following script creates three layer sets, then nests a second layer set in each layer set, and then creates a text layer in each nested set that that displays the text "Layer in *n* Set Inside *n* Set", where *n* represents the ordinal number of the set (first, second, or third).

Note: This script uses the ExtendScript `$` debugging object. For further details, see the *JavaScript Tools Guide*.

LayerSets.jsx

```
$.level = 1

//close all open documents
while (app.documents.length) {
```

```
   app.activeDocument.close()
}

// create a working document
var docRef = app.documents.add()

// create an array to hold the layer sets
var myLayerSets = new Array()

// Create an array to hold the text
var textArray = Array("First", "Second", "Third")

//Create an indexer variable
var i = 0

// Create three layer sets at the top level
for (i = 0; i < 3; i++) {
   myLayerSets[i] = new Array()
   myLayerSets[i][0] = docRef.layerSets.add()
}

// Rearrange the layer sets with the first one on top, second next, etc.
myLayerSets[1][0].moveAfter(myLayerSets[0][0])
myLayerSets[2][0].moveAfter(myLayerSets[1][0])

// Create a layer set inside each layer set
for (i = 0; i < 3; i++) {
   myLayerSets[i][0].name = textArray[i] + " Set"
   myLayerSets[i][1] = myLayerSets[i][0].layerSets.add()
   myLayerSets[i][1].name = "Inside " + textArray[i] + " Set"
}

// Create an array to hold the layers
var myLayers = new Array()

// Create a text layer with a description inside each layer set
for (i = 0; i < 3; i++) {
   myLayers[i] = myLayerSets[i][1].artLayers.add()
   myLayers[i].kind = LayerKind.TEXT
   myLayers[i].textItem.contents = "Layer in " + textArray[i] + " Set Inside "
          + textArray[i] + " Set"
   myLayers[i].textItem.position = Array(app.activeDocument.width * i * 0.33,
          app.activeDocument.height * (i + 1) * 0.25)
   myLayers[i].textItem.size = 12
}
```

MeasurementLog

The measurement log for the application. Access through the Application.measurementLog property.

Note: This feature is available in the Extended Version only.

Methods

Method	Parameter type	Returns	What it does
exportMeasurements ([file] [, range]) [, dataPoints])	File MeasurementRange array of string		Export measurement to a file.
deleteMeasurements ([range])	MeasurementRange		Delete measurements from the log.

MeasurementScale

The measurement scale for the document. Access through the Document.measurementScale property. For example:

```
app.activeDocument.measurementScale.pixelLength = 25
```

Note: This feature is available in the Extended Version only.

Properties

Property	Value type	What it is
pixelLength	number	Read-write. The length in pixels this scale equates to.
logicalLength	number	Read-write. The logical length this scale equates to.
logicalUnits	string	Read-write. The logical units for this scale.

NoColor

Represents a missing color object, used in the SolidColor object.

See also CMYKColor, GrayColor, HSBColor, LabColor, RGBColor

Properties

Property	Value type	What it is
`typename`	`string`	Read-only. The class name of the referenced `noColor` object.

Notifier

An event-handler object that tells a script to execute specified code when a specified event occurs. Notifiers must be enabled using the Application.notifiersEnabled property.

Access through the Application.notifiers collection.

Note: Events that occur within scripts do not generally trigger notifiers, because they occur inside a "play script" event.

Properties

Property	Value type	What it is
`event`	`string`	Read-only. The event identifier, a four-character code or a unique string. For a list of four-character codes, see Appendix A: Event ID Codes.
`eventClass`	`string`	Read-only. The class identifier, a four-character code or a unique string. When an event applies to multiple types of objects, use this propery to distinguish which object this notifier applies to. For example, the Make event (`"Mk  "`) can apply to documents (`"Dcmn"`), channels (`"Chnl"`) and other objects.
`eventFile`	`File`	Read-only. The path to the file to execute when the event occurs and activates the notifier.
`parent`	`Application`	Read-only. The containing application.
`typename`	`string`	Read-only. The class name of the referenced object.

Methods

Method	Parameter type	Returns	What it does
`remove ()`			Deletes this object. You can also remove a `Notifier` object from the Script Events Manager drop-down list by deleting the file named `Script Events Manager.xml` from the Photoshop preferences folder. See Adobe Photoshop CS6 help for more information.

Notifiers

The collection of Notifier objects in the document. Access through the Application.notifiers collection property. For example:

```
var notRef = app.notifiers.add("OnClickGoButton", eventFile)
```

Notifiers must be enabled using the Application.notifiersEnabled property.

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the `notifiers` collection.
parent	Application	Read-only. The `notifiers` object's container
typename	string	Read-only. The class name of the referenced `notifiers` object.

Methods

Method	Parameter type	Returns	What it does
add (event, eventFile [, eventClass])	string File string	Notifier	Creates a notifier object and adds it to this collection. `event` defines the class ID of the event: use a 4-characters code or a unique string. See Appendix A: Event ID Codes. `eventFile` defines the script file that executes when the event occurs. When an event applies to multiple types of objects, use the `eventClass` (a 4-character ID or unique string) to distinguish which object this `Notifier` applies to. For example, the Make event (`"Mk  "`) applies to documents (`"Dcmn"`), channels (`"Chnl"`) and other objects. **Tip:** When specifying an event or event calss wtih a 4-character ID code, omit the single quotes in your code.
removeAll ()			Removes all member objects from the `notifiers` collection. You can also remove a `notifier` object from the Script Events Manager drop-down list by deleting the file named `Script Events Manager.xml` from the Photoshop preferences folder. See Adobe Photoshop CS6 help for more information.

PathItem

A path or drawing object, such as the outline of a shape or a straight or curved line, which contains sub paths that define its geometry.

Access through the collection in the Document.pathItems property. For example, this selects a named path item:

```
var currentPathItem = app.activeDocument.pathItems.getByName("myPath");
currentPathItem.select()
```

Create these objects by passing a set of SubPathInfo objects to the PathItems.add() method. This method creates a SubPathItem object for each SubPathInfo object, and creates and returns a new PathItem object for the path represented by all of the subpaths.

Properties

Property	Value type	What it is
kind	PathKind	Read-write. The type.
name	string	Read-write. The name.
parent	Document	Read-only. The containing document.
subPathItems	SubPathItems	Read-only. The contained sub-path objects.
typename	string	Read-only. The class name of the referenced pathItem object.

Methods

Method	Parameter type	Returns	What it does
deselect ()			Deselects this pathItem object.
duplicate (name)	string		Duplicates this pathItem object with the new name.
fillPath ([fillColor] [, mode] [, opacity] [, preserveTransparency] [, feather] [, wholePath] [, antiAlias])	SolidColor ColorBlendMode number [0..100] boolean number [0.0..250.0] boolean boolean		Fills the area enclosed by this path. opacity is a percentage. feather is in pixels. If wholePath is true, all subpaths are used when doing the fill (default: true).

Method	Parameter type	Returns	What it does
makeClippingPath ([flatness])	number [0.2..100]		Makes this the clipping path for this document. flatness tells the PostScript printer how to approximate curves in the path.
makeSelection ([feather] [, antiAlias] [, operation])	number [0.0..250.0] boolean SelectionType		Makes a Selection object whose border is this path. feather is in pixels.
remove ()			Deletes this object.
select ()			Makes this the active or selected PathItem object.
strokePath ([tool] [, simulatePressure])	ToolType boolean		Strokes the path with the specified tool.

PathItem sample script

The following creates a path in three segments: two diagonal lines that form a *V*, and a curved line above the *V* that makes it look like a 2D ice cream cone.

Paths.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// first close all the open documents
while (app.documents.length) {
   app.activeDocument.close()
   }

// create a document to work with
var docRef = app.documents.add(5000, 7000, 72, "Simple Line")

//line 1--it's a straight line so the coordinates for anchor, left, and right
//for each point have the same coordinates
var lineArray = new Array()
   lineArray[0] = new PathPointInfo
   lineArray[0].kind = PointKind.CORNERPOINT
   lineArray[0].anchor = Array(100, 100)
```

```
   lineArray[0].leftDirection = lineArray[0].anchor
   lineArray[0].rightDirection = lineArray[0].anchor

   lineArray[1] = new PathPointInfo
   lineArray[1].kind = PointKind.CORNERPOINT
   lineArray[1].anchor = Array(150, 200)
   lineArray[1].leftDirection = lineArray[1].anchor
   lineArray[1].rightDirection = lineArray[1].anchor

var lineSubPathArray = new Array()
   lineSubPathArray[0] = new SubPathInfo()
   lineSubPathArray[0].operation = ShapeOperation.SHAPEXOR
   lineSubPathArray[0].closed = false
   lineSubPathArray[0].entireSubPath = lineArray

// line 2
var lineArray2 = new Array()
   lineArray2[0] = new PathPointInfo
   lineArray2[0].kind = PointKind.CORNERPOINT
   lineArray2[0].anchor = Array(150, 200)
   lineArray2[0].leftDirection = lineArray2[0].anchor
   lineArray2[0].rightDirection = lineArray2[0].anchor

   lineArray2[1] = new PathPointInfo
   lineArray2[1].kind = PointKind.CORNERPOINT
   lineArray2[1].anchor = Array(200, 100)
   lineArray2[1].leftDirection = lineArray2[1].anchor
   lineArray2[1].rightDirection = lineArray2[1].anchor

   lineSubPathArray[1] = new SubPathInfo()
   lineSubPathArray[1].operation = ShapeOperation.SHAPEXOR
   lineSubPathArray[1].closed = false
   lineSubPathArray[1].entireSubPath = lineArray2

//ice cream curve
//it’s a curved line, so there are 3 points, not 2
//coordinates for the middle point (lineArray3[1]) are different.
//The left direction is positioned "above" the anchor on the screen.
//The right direction is positioned "below" the anchor
//You can change the coordinates for these points to see
//how the curve works...
var lineArray3 = new Array()
   lineArray3[0] = new PathPointInfo
   lineArray3[0].kind = PointKind.CORNERPOINT
   lineArray3[0].anchor = Array(200, 100)
   lineArray3[0].leftDirection = lineArray3[0].anchor
   lineArray3[0].rightDirection = lineArray3[0].anchor

   lineArray3[1] = new PathPointInfo
   lineArray3[1].kind = PointKind.CORNERPOINT
   lineArray3[1].anchor = Array(150, 50)
   lineArray3[1].leftDirection = Array(100, 50)
   lineArray3[1].rightDirection = Array(200, 50)

   lineArray3[2] = new PathPointInfo
   lineArray3[2].kind = PointKind.CORNERPOINT
   lineArray3[2].anchor = Array(100, 100)
   lineArray3[2].leftDirection = lineArray3[2].anchor
   lineArray3[2].rightDirection = lineArray3[2].anchor
```

```
    lineSubPathArray[2] = new SubPathInfo()
    lineSubPathArray[2].operation = ShapeOperation.SHAPEXOR
    lineSubPathArray[2].closed = false
    lineSubPathArray[2].entireSubPath = lineArray3

//create the path item
var myPathItem = docRef.pathItems.add("A Line", lineSubPathArray)

// stroke it so we can see something
myPathItem.strokePath(ToolType.BRUSH)

// Reset the application preferences
preferences.rulerUnits = startRulerUnits
preferences.typeUnits = startTypeUnits
displayDialogs = startDisplayDialogs
```

PathItems

The collection of PathItem objects in a document.

Access through the Document.pathItems collection property. For example, this creates a new path item using a previously-defined set of subpaths:

```
app.activeDocument.pathItems.add("myPath", lineSubPathInfoArray);
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of `pathItem` objects in the `pathItems` collection.
parent	Document	Read-only. The `pathItems` object's container.
typename	string	Read-only. The class name of the referenced `pathItems` object.

Methods

Method	Parameter type	Returns	What it does
add (name, entirePath)	string array of SubPathInfo	PathItem	Creates a new path item object and adds it to this collection. A new SubPathItem object is created for each `SubPathInfo` object provided in `entirePath`, and those `SubPathItem` objects are added to the subPathItems collection of the returned PathItem.
getByName (name)	string	PathItem	Get the first element in the `pathItems` collection with the provided name.
removeAll ()			Removes all `pathItem` objects from the `pathItems` collection.

PathPoint

Represents the anchor and control-handle endpoints for a path segment. Each point (the anchor point, left-direction point, and right-direction point) is an array containing X and Y position coordinates.

- Use the `PathPoint` object to retrieve information about the points that describe existing path segments. The properties are read-only. Access PathPoint objects through the SubPathItem.pathPoints property.
- Use PathPointInfo with PathItems.add() to create path points. The properties are writeable.

For paths that are straight segments (not curved), the coordinates of all three points are the same. For curved segements, the the coordinates are different. The difference between the anchor point and the left or right direction points determines the arc of the curve. You use the left direction point to bend the curve "outward" or make it convex; you use the right direction point to bend the curve "inward" or make it concave.

Properties

Property	Value type	What it is
anchor	array of number	Read-only. The X and Y coordinates of the anchor point of the curve.
kind	PointKind	Read-only. The role (corner or smooth) this point plays in the containing path segment.
leftDirection	array of number	Read-only. The location of the left-direction endpoint ('in' position).
parent	SubPathItem	Read-only. The containing subpath object.
rightDirection	array of number	Read-only. The location of the right-direction endpoint ('out' position).
typename	string	Read-only. The class name of the referenced `PathPoint` object.

PathPointInfo

Used to create a PathPoint, which represents the anchor and control-handle endpoints for a path segment. Each point (the anchor point, left-direction point, and right-direction point) is an array containing X and Y position coordinates.

- Use the JavaScript `new` operator to create these objects, and store them in the SubPathInfo.entireSubPath property before using that object to create a path item with PathItems.add().

  See the PathPointInfo sample script below.

- The resulting SubPathItem objects contain the resulting PathPoint objects. Use the PathPoint object to retrieve information about the points that describe existing path segments. The properties are read-only.

For paths that are straight segments (not curved), the coordinates of all three points are the same. For curved segements, the the coordinates are different. The difference between the anchor point and the left or right direction points determines the arc of the curve. You use the left direction point to bend the curve "outward" or make it convex; you use the right direction point to bend the curve "inward" or make it concave.

Properties

Property	Value type	What it is
anchor	array of number	Read-write. The X and Y coordinates of the anchor point of the curve.
kind	PointKind	Read-write. The role (corner or smooth) this point plays in the containing path segment.
leftDirection	array of number	Read-write. The location of the left-direction endpoint ('in' position).
rightDirection	array of number	Read-write. The location of the right-direction endpoint ('out' position).
typename	string	Read-only. The class name of the referenced `PathPointInfo` object.

PathPointInfo sample script

```
function drawLine(doc, start, stop) {

    var startPoint = new PathPointInfo();
    startPoint.anchor = start;
    startPoint.leftDirection = start;
    startPoint.rightDirection = start;
    startPoint.kind = PointKind.CORNERPOINT;

    var stopPoint = new PathPointInfo();
    stopPoint.anchor = stop;
    stopPoint.leftDirection = stop;
    stopPoint.rightDirection = stop;
    stopPoint.kind = PointKind.CORNERPOINT;
```

```
    var spi = new SubPathInfo();
    spi.closed = false;
    spi.operation = ShapeOperation.SHAPEXOR;
    spi.entireSubPath = [startPoint, stopPoint];

    var line = doc.pathItems.add("Line", [spi]);
    line.strokePath(ToolType.PENCIL);
    line.remove();
    };

drawLine(app.activeDocument, [100,100], [200,200]);
```

PathPoints

A collection of PathPoint objects that define a subpath, kept in the SubPathItem.pathPoints property.

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the collection.
parent	SubPathItem	Read-only. The containing subpath object.
typename	string	Read-only. The class name of the referenced PathPoints object.

PDFOpenOptions

Options for opening a document in generic Adobe PDF format using the Application.open() method.

Properties

Property	Value type	What it is
antiAlias	boolean	Read-write. True to use antialias.
bitsPerChannel	BitsPerChannelType	Read-write. The number of bits per channel.
constrainProportions	boolean	DEPRECATED for Adobe Photoshop CS6.
cropPage	CropToType	Read-write. The method of cropping to use.
height	UnitValue	DEPRECATED for Adobe Photoshop CS6.
mode	OpenDocumentMode	Read-write. The color model to use.
name	string	Read-write. The name of the object.
page	number	Read-write. The page or image to which to open the document, depending on the value of usePageNumber.
resolution	number	Read-write. The resolution of the document (in pixels per inch).
suppressWarnings	boolean	Read-write. True to suppress warnings when opening the document.
typename	string	Read-only. The class name of the referenced PDFOpenOptions object.
usePageNumber	boolean	Read-write. When true, the page property refers to a page number; when false, it refers to an image number.
width	UnitValue	DEPRECATED for Adobe Photoshop CS6.

PDFSaveOptions

Options for saving a document in Adobe PDF format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels with the file.
annotations	boolean	Read-write. True to save comments with the file.
colorConversion	boolean	Read-write. True to convert the color profile to a destination profile.
convertToEightBit	boolean	Read-write. True to convert a 16-bit image to 8-bit for better compatibility with other applications.
description	string	Read-write. Description of the save options to use.
destinationProfile	string	Read-write. Description of the final RGB or CMYK output device, such as a monitor or a press standard.
downgradeColorProfile	boolean	DEPRECATED for Adobe Photoshop CS6.
downSample	PDFResample	Read-write. The down sample method to use.
downSampleSize	number	Read-write. The size to downsample images if they exceed the limit in pixels per inch.
downSampleSizeLimit	number	Read-write. Limits downsampling or subsampling to images that exceed this value in pixels per inch.
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
embedFonts	boolean	DEPRECATED for Adobe Photoshop CS6.
embedThumbnail	boolean	Read-write. True to include a small preview image in Adobe PDF files.
encoding	PDFEncoding	Read-write. The type of compression to use (default: `PDFEncoding.PDFZIP`).
interpolation	boolean	DEPRECATED for Adobe Photoshop CS6.

Property	Value type	What it is (Continued)
jpegQuality	number [0..12]	Read-write. The quality of the produced image, which is inversely proportionate to the compression amount. Valid only when encoding = PDFEncoding.JPEG.
layers	boolean	Read-write. True to save the document's layers.
optimizeForWeb	boolean	Read-write. True to improve performance of PDF files on Web servers.
outputCondition	string	Read-write. An optional comment field for inserting descriptions of the output condition. The text is stored in the PDF/X file.
outputConditionID	string	Read-write. Indentifier for the output condition.
PDFCompatibility	PDFCompatibility	Read-write. The PDF version to make the document compatible with.
PDFStandard	PDFStandard	Read-write. The PDF standard to make the document compatible with.
preserveEditing	boolean	Read-write. True to reopen the PDF in Adobe Photoshop CS6 with native Photoshop data intact.
presetFile	string	Read-write. The preset file to use for settings. **Note:** This option overrides other settings.
profileInclusionPolicy	boolean	Read-write. True to show which profiles to include.
registryName	string	Read-write. URL where the output condition is registered.
spotColors	boolean	Read-write. True to save spot colors.
tileSize	nunber	Read-write. Compression option. Valid only when encoding = PDFEncoding.JPEG2000.
transparency	boolean	DEPRECATED for Adobe Photoshop CS6.
typename	string	Read-only. The class name of the referenced PDFSaveOptions object.
useOutlines	boolean	DEPRECATED for Adobe Photoshop CS6.

Property	Value type	What it is (Continued)
vectorData	boolean	DEPRECATED for Adobe Photoshop CS6.
view	boolean	Read-write. True to open the saved PDF in Adobe Acrobat.

PhotoCDOpenOptions

DEPRECATED in Adobe Photoshop CS6. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS6 Install DVD.

Options for opening a document in Kodak Photo CD (PCD) format (including high-resolution files from Pro Photo CD discs) using the Application.open() method.

Properties

Property	Value type	What it is
colorProfileName	string	Read-write. The profile to use when reading the image.
colorSpace	PhotoCDColorSpace	Read-write. The colorspace for the image.
orientation	Orientation	Read-write. The image orientation.
pixelSize	PhotoCDSize	Read-write. The image dimensions.
resolution	number	Read-write. The image resolution (in pixels per inch).
typename	string	Read-only. The class name of the referenced photoCDOpenOptions object.

PhotoshopSaveOptions

Options for saving a document in PSD format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
annotations	boolean	Read-write. True to save the annotations.
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
layers	boolean	Read-write. True to preserve the layers.
spotColors	boolean	Read-write. True to save the spot colors.
typename	string	Read-only. The class name of the referenced photoshopSaveOptions object.

PICTFileSaveOptions

Options for saving a document in PICT format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
compression	PICTCompression	Read-write. The type of compression to use (default: PICTCompression.NONE).
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
resolution	PICTBitsPerPixels	Read-write. The number of bits per pixel.
typename	string	Read-only. The class name of the referenced PICTFileSaveOptions object.

PICTResourceSaveOptions

Options for saving a document as a PICT Resource file using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
compression	PICTCompression	Read-write. The type of compression to use (default: PICTCompression.NONE).
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
name	string	Read-write. The name of the PICT resource.
resolution	PICTBitsPerPixels	Read-write. The number of bits per pixel.
resourceID	number	Read-write. The ID of the PICT resource (default: 128).
typename	string	Read-only. The class name of the referenced PICTResourceSaveOptions object.

PicturePackageOptions

Options for a picture package created with Application.makePicturePackage().

Properties

Property	Value type	What it is
content	PicturePackageTextType	Read-write. The content information (default: PicturePackageTextType.NONE).
flatten	boolean	Read-write. True if all layers in the final document are flattened (default: true).
font	GalleryFontType	Read-write. The font used for security text (default: GalleryFontType.ARIAL).
fontSize	number	Read-write. The font size used for security text (default: 12).
layout	string	Read-write. The layout to use to generate the picture package (default: "(2)5x7").
mode	NewDocumentMode	Read-write. Read-write. The color profile to use as the document mode (default: NewDocumentMode.RGB).
opacity	number	Read-write. The web page security opacity as a percent (default: 100).
resolution	number	Read-write. The resolution of the document in pixels per inch (default: 72.0).
text	string	Read-write. The picture package custom text. Valid only when content = PicturePackageType.USER.
textColor	RGBColor	Read-write. The color to use for security text.
textPosition	GallerySecurityTextPositionType	Read-write. The security text position (default: GallerySecurityTextPositionType.CENTERED).
textRotate	GallerySecurityTextRotateType	Read-write. The orientation to use for security text (default: GallerySecurityTextRotateType.ZERO).
typename	string	Read-only. The class name of the referenced PicturePackageOptions object.

PixarSaveOptions

Options for saving a document in Pixar format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
typename	string	Read-only. The class name of the referenced `PixarSaveOptions` object.

PNGSaveOptions

Options for saving a document in PNG format using the Document.saveAs() method.

Properties

Property	Value type	What it is
compression	number [0..9]	Read-write. The compression value (default: 0).
interlaced	boolean	Read-write. True to interlace rows (default: false).
typename	string	Read-only. The class name of the referenced PNGSaveOptions object.

Preferences

Represents application preferences for Photoshop. Access this object through the Application.preferences property. For example:

```
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
```

Setting values in this object is equivalent to selecting **Edit > Preferences** (in Windows) or **Photoshop > Preferences** (in Mac OS) in the Adobe Photoshop CS6 application. For explanations of individual settings, see Adobe Photoshop CS6 Help.

Properties

Property	Value type	What it is
additionalPluginFolder	File	Read-write. The path to an additional plug-in folder. Valid only when useAdditionalPluginFolder = true.
appendExtension	SaveBehavior	Read-write. The preferred policy for writing file extensions in Windows.
askBeforeSavingLayeredTIFF	boolean	Read-write. True to ask the user to verify layer preservation options when saving a file in TIFF format.
autoUpdateOpenDocuments	boolean	Read-write. True to automatically update open documents.
beepWhenDone	boolean	Read-write. True to beep when a process finishes.
colorChannelsInColor	boolean	Read-write. True to display component channels in the Channels palette in color.
colorPicker	ColorPicker	Read-write. The preferred color selection tool.
columnGutter	number [0.1..600.0]	Read-write. The width of the column gutters (in points).
columnWidth	number [0.1..600.0]	Read-write. Column width (in points)
createFirstSnapshot	boolean	Read-write. True to automatically make the first snapshot when a new document is created.
dynamicColorSliders	boolean	Read-write. True if dynamic color sliders appear in the Color palette.
editLogItems	EditLogItemsType	Read-write. The preferred level of detail in the history log. Valid only when useHistoryLog = true.

Property	Value type	What it is (Continued)
exportClipboard	boolean	Read-write. True to retain Adobe Photoshop CS6 contents on the clipboard after you exit the application.
fontPreviewSize	FontPreviewType	Read-write. The preferred type size to use for font previews in the type tool font menus.
fullSizePreview	boolean	Read-write. True to show image preview as a full size image, false to show thumbnail (in Mac OS only).
gamutWarningOpacity	number [0..100]	Read-write. Opacity value as a percentage.
gridSize	GridSize	Read-write. The preferred size to use for squares in the grid.
gridStyle	GridLineStyle	Read-write. The preferred formatting style for non-printing grid lines.
gridSubDivisions	number [1..100]	Read-write. Number of grid subdivisions.
guideStyle	GuideLineStyle	Read-write. The preferred formatting style for non-printing guide lines.
iconPreview	boolean	Read-write. True to use icon previews (in Mac OS only).
imageCacheLevels	number [1..8]	Read-write. The number of images to hold in the cache.
imagePreviews	SaveBehavior	Read-write. The preferred policy for writing image previews in Windows.
interpolation	ResampleMethod	Read-write. The method to use to assign color values to any new pixels created when an image is resampled or resized.
keyboardZoomResizesWindows	boolean	Read-write. True to automatically resize the window when zooming in or out using keyboard shortcuts.
macOSThumbnail	boolean	Read-write. True to create a thumbnail when saving the image (in Mac OS only).
maximizeCompatibility	QueryStateType	Read-write. The preferred policy for checking whether to maximize compatibility when opening PSD files.
maxRAMuse	number [5..100]	Read-write. The maximum percentage of available RAM used by Adobe Photoshop CS6 (5 - 100).
nonLinearHistory	boolean	Read-write. True to allow non-linear history.

Property	Value type	What it is (Continued)
numberofHistoryStates	number	Read-write. The number of history states to preserve.
otherCursors	OtherPaintingCursors	Read-write. The preferred type of pointer to use with certain tools.
paintingCursors	PaintingCursors	Read-write. The preferred type of pointer to use with certain tools.
parent	Application	Read-write. The containing application.
pixelDoubling	boolean	Read-write. True to halve the resolution (double the size of pixels) to make previews display more quickly.
pointSize	PointType	Read-write. The point/pica size.
recentFileListLength	number [0..30]	Read-write. The number of items in the recent file list.
rulerUnits	Units	Read-write. The unit the scripting system will use when receiving and returning values.
saveLogItems	SaveLogItemsType	Read-write. Thepreferred location of history log data when saving the history items.
saveLogItemsFile	File	Read-write. The path to the history log file, when the preferred location is a file.
savePaletteLocations	boolean	Read-write. True to make new palette locations the default location.
showAsianTextOptions	boolean	Read-write. True to display Asian text options in the Paragraph palette.
showEnglishFontNames	boolean	Read-write. True to list Asian font names in English.
showSliceNumber	boolean	Read-write. True to display slice numbers in the document window when using the Slice tool.
showToolTips	boolean	Read-write. True to show pop up definitions on mouse over.
smartQuotes	boolean	Read-write. True to use curly, false to use straight quote marks.
textFontSize	FontSize	Read-write. Size of the small font used in panels and dialogs.
typename	string	Read-only. The class name of the referenced preferences object.

Property	Value type	What it is (Continued)
typeUnits	TypeUnits	Read-write. The preferred unit for text character measurements.
useAdditionalPluginFolder	boolean	Read-write. True to use an additional folder for compatible plug-ins stored with a different application.
useHistoryLog	boolean	Read-write. True to create a log file for history states.
useLowerCaseExtension	boolean	Read-write. True to use lowercase for file extensions.
useShiftKeyForToolSwitch	boolean	Read-write. True to enable cycling through a set of hidden tools.
useVideoAlpha	boolean	Read-write. True to enable Adobe Photoshop CS6 to send transparency information to your computer's video board. (Requires hardware support.)
windowsThumbnail	boolean	Read-write. True to create a thumbnail when saving the image in Windows. (Requires hardware support.)

PresentationOptions

Options for Adobe PDF presentations created using Application.makePDFPresentation().

Properties

Property	Value type	What it is
autoAdvance	boolean	Read-write. True to auto advance images when when viewing the presentation (default: `true`). Valid only when presentation `= true`.
includeFilename	boolean	Read-write. True to include the file name for the image (default: `false`).
interval	number [1..60]	Read-write. The time in seconds before the view is auto advanced (default: `5`). Valid only when autoAdvance `= true`.
loop	boolean	Read-write. True to begin the presentation again after the last page (default: `false`). Valid only when autoAdvance `= true`.
magnification	MagnificationType	Read-write. The magnification type to use when viewing the image.
PDFFileOptions	PDFSaveOptions	Read-write. Options to use when creating the PDF file.
presentation	boolean	Read-write. True if the output will be a presentation (default: `false`); when `false`, the output is a Multi-Page document.
transition	TransitionType	Read-write. The method for transition from one image to the next (default: `TransitionType.NONE`). Valid only when autoAdvance `= true.`.
typename	string	Read-only. The class name of the referenced `PresentationOptions` object.

RawFormatOpenOptions

Options for opening a document in RAW format using the Application.open() method.

Properties

Property	Value type	What it is
bitsPerChannel	number	Read-write. The number of bits for each channel. The only valid values are `BitsPerChannelType.EIGHT` or `BitsPerChannelType.SIXTEEN`.
byteOrder	ByteOrder	Read-write. The order in which multibyte values are read. Valid only when bitsPerChannel = `BitsPerChannelType.SIXTEEN`.
channelNumber	number [1..56]	Read-write. The number of channels in the image. The value of cannot exceed the number of channels in the image. When bitsPerChannel = `BitsPerChannelType.SIXTEEN`, the only valid values are 1, 3, or 4.
headerSize	number [0..1919999]	Read-write. The number of bytes of information that will appear in the file before actual image information begins; that is, the number of zeroes inserted at the beginning of the file as placeholders.
height	number	Read-write. The height of the image (in pixels).
interleaveChannels	boolean	Read-write. True to store color values sequentially.
retainHeader	boolean	Read-write. True to retain the header when saving. Valid only when headerSize is 1 or greater.
typename	string	Read-only. The class name of the referenced `RawFormatOpenOptions` object.
width	number	Read-write. The image width in pixels.

RawSaveOptions

Options for saving a document in RAW format using the Document.saveAs() method.

Properties

Property	Value type	What it is
`alphaChannels`	`boolean`	Read-write. True if alpha channels should be saved.
`spotColors`	`boolean`	Read-write. True if the spot colors should be saved.
`typename`	`string`	Read-only. The class name of the referenced `RawSaveOptions` object.

RGBColor

Defines an RGB color, used in the SolidColor object.

See also CMYKColor, GrayColor, HSBColor, LabColor, NoColor.

Properties

Property	Value type	What it is
blue	number [0..255]	Read-write. The blue color value (default: 255).
green	number [0..255]	Read-write. The green color value (default: 255)
hexValue	string	Read-write. The hexadecimal representation of the color.
red	number [0..255]	Read-write. The red color value (default: 255)
typename	string	Read-only. The class name of the referenced `RGBColor` object.

Selection

The selected area of a document or layer. Access through the Document.selection property. For example:

```
app.activeDocument.selection.fill(app.foregroundColor)
```

Many of the properties and methods use the UnitValue type, which combines measurement values with the measurement unit. For information about this type, see the *JavaScript Tools Guide*.

Properties

Property	Value type	What it is
bounds	array of UnitValue	Read-only. The bounding rectangle of the entire selection.
parent	Document	Read-only. The object's container.
solid	boolean	Read-only. True if the bounding rectangle is a solid.
typename	string	Read-only. The class name of the referenced selection object.

Methods

Method	Parameter type	Returns	What it does
clear ()			Clears the selection and does not copy it to the clipboard.
contract (by)	UnitValue		Contracts (reduces) the selection by the specified amount.
copy ([merge])	boolean		Copies the selection to the clipboard. When the optional argument is used and set to true, a merged copy is performed (all visible layers in the selection are copied).
cut ()			Clears the current selection and copies it to the clipboard.
deselect ()			Deselects the current selection.
expand (by)	UnitValue		Expands the selection by the specified amount.
feather (by)	UnitValue		Feathers the edges of the selection by the specified amount.

Method	Parameter type	Returns	What it does (Continued)
fill (filltype [, mode] [, opacity] [, preserveTransparency])	SolidColor ColorBlendMode number [1..100] boolean		Fills the selection. opacity is a percentage value.
grow (tolerance, antiAlias)	number boolean		Grows the selection to include all adjacent pixels falling within the specified tolerance range.
invert ()			Inverts the selection (deselects the selection and selects the rest of the layer or document). **Tip:** To flip the selection shape, see rotate.
load (from [, combination] [, inverting])	Channel SelectionType boolean		Loads the selection from the specified channel.
makeWorkPath ([tolerance])	number		Makes this selection item the work path for this document.
resize ([horizontal] [, vertical] [, anchor])	number number AnchorPosition		Resizes the selected area to the specified dimensions and anchor position.
resizeBoundary ([horizontal] [, vertical] [, anchor])	number number AnchorPosition		Changes the size of the selection to the specified dimensions around the specified anchor.
rotate (angle [, anchor])	number AnchorPosition		Rotates the selection by the specified amount around the specified anchor point.
rotateBoundary (angle [, anchor])	number AnchorPosition		Rotates the boundary of the selection around the specified anchor.
select (region [, type] [, feather] [, antiAlias])	array of number SelectionType number boolean		Selects the specified region. The region parameter is an array of four coordinates, [left, top, right, bottom].
selectAll ()			Selects the entire layer.

Method	Parameter type	Returns	What it does (Continued)
selectBorder (width)	UnitValue		Selects the selection border only (in the specified width); subsequent actions do not affect the selected area within the borders.
similar (tolerance, antiAlias)	number boolean		Grows the selection to include pixels throughout the image falling within the tolerance range.
smooth (radius)	number		Cleans up stray pixels left inside or outside a color-based selection (within the radius specified in pixels).
store (into [, combination])	Channel SelectionType		Saves the selection as a channel.
stroke (strokeColor, width [, location] [, mode] [, opacity] [, preserveTransparency])	SolidColor number StrokeLocation ColorBlendMode number [1..100] boolean		Strokes the selection border. `opacity` is a percentage value.
translate ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the entire selection relative to its current position.
translateBoundary ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the selection relative to its current position.

Selection sample script

The following script creates a checkerboard using the following steps:

- Create an 800 x 800 pixel document.
- Divide the entire document into 100 x 100 pixel squares.
- Select every other square in the first row, then shift the selection criteria to select the alternate squares in the following row. Repeat until every other square in the document is selected.
- Fill the selected squares with the foreground color from the palette.
- Invert the selection and fill the newly selected squares with the background color from the palette.
- Deselect the squares to remove the selection outlines (the "marching ants").

Selection.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
```

```
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS6 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

//Close all the open documents
while (app.documents.length) {
   app.activeDocument.close()
}

//Create variables for the 800 pixel board divided in even 100 x 100 squares
var docSize = 800
var cells = 8
var cellSize = docSize / cells

// create a new document
var checkersDoc = app.documents.add(docSize, docSize, 72, "Checkers")

// Create a variable to use for selecting the checker board
// That allows me to shift the selection one square to the right
//on every other row, and then shift back for the rows in between.
var shiftIt = true

// loop through vertically to create the first row
for (var v = 0; v < docSize; v += cellSize) {

   // Switch the shift for a new row
   shiftIt = !shiftIt

   // loop through horizontally
   for (var h = 0; h < docSize; h += (cellSize * 2)) {

          // push over the cellSize to start with only
          if (shiftIt && h == 0) {
             h += cellSize
      }

          // Select a square
          selRegion = Array(Array(h, v),
                      Array(h + cellSize, v),
                      Array(h + cellSize, v + cellSize),
                      Array(h, v + cellSize),
                      Array(h, v))

          // In the first ineration of the loop, start the selection
          //In subsequent iterations, use the EXTEND constant value
          //of the select() method to add to the selection (in the loop's else clause)
          if (h == 0 && v == 0) {
             checkersDoc.selection.select(selRegion)
          } else {
             checkersDoc.selection.select(selRegion, SelectionType.EXTEND)
      }

          // turn this off for faster execution
          // turn this on for debugging
          WaitForRedraw()
   }
```

```
}

// Fill the current selection with the foreground color
checkersDoc.selection.fill(app.foregroundColor)

//Invert the selection
checkersDoc.selection.invert()

// Fill the new selection with the background color
checkersDoc.selection.fill(app.backgroundColor)

// Clear the selection to get rid of the non-printing borders
checkersDoc.selection.deselect()

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs

// A helper function for debugging
// It also helps the user see what is going on
// if you turn it off for this example you
// get a flashing cursor for a number time
function WaitForRedraw()
{
   var eventWait = charIDToTypeID("Wait")
   var enumRedrawComplete = charIDToTypeID("RdCm")
   var typeState = charIDToTypeID("Stte")
   var keyState = charIDToTypeID("Stte")

   var desc = new ActionDescriptor()

   desc.putEnumerated(keyState, typeState, enumRedrawComplete)

   executeAction(eventWait, desc, DialogModes.NO)
}
```

SGIRGBSaveOptions

Options for saving a document in SGIRGB format using the Document.saveAs() method.

Note: The SGIRGB format is not installed automatically with Adobe Photoshop CS6.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
spotColors	boolean	Read-write. True to save the spot colors.
typename	string	Read-only. The class name of the referenced SGIRGBSaveOptions object.

SolidColor

A color definition used in the document. Maps a color to equivalents in all available color models.

- Used in Application.backgroundColor and foregroundColor properties, in Channel.color, in ColorSampler.color, and in TextItem.color
- Passed to PathItem.fillPath(), Selection.fill(), and Selection.stroke().

Properties

Property	Value type	What it is
cmyk	CMYKColor	Read-write. The CMYK color mode.
gray	GrayColor	Read-write. The Grayscale color mode.
hsb	HSBColor	Read-write. The HSB color mode.
lab	LabColor	Read-write. The LAB color mode.
model	ColorModel	Read-write. The color model.
nearestWebColor	RGBColor	Read-only. The nearest web color to the current color.
rgb	RGBColor	Read-write. The RGB color mode.
typename	string	Read-only. The class name of the referenced `SolidColor` object.

Methods

Method	Parameter type	Returns	What it does
isEqual (color)	SolidColor	boolean	True if the `SolidColor` object is visually equal to the specified color.

SubPathInfo

An array of PathPoint objects that describes a straight or curved segment of a path, used to create a SubPathItem.

Pass an array of these objects to the PathItems.add() method. This method creates a SubPathItem object for each `SubPathInfo` object, and creates and returns a new PathItem object for the path represented by all of the subpaths.

- Use `SubPathInfo` to create subpaths; the properties are writeable.
- Use the SubPathItem object to retrieve information about existing subpaths. The properties are read-only.

Properties

Property	Value type	What it is
`closed`	`boolean`	Read-write. True if the path describes an enclosed area.
`entireSubPath`	`array of PathPoint`	Read-write.
`operation`	`ShapeOperation`	Read-write. The subpath's operation on other subpaths. Specifies how to combine the shapes if the destination path already has a selection.
`typename`	`string`	Read-only. The class name of the referenced `SubPathInfo` object.

SubPathItem

Represents a subpath; a collection of subpaths make up a PathItem.

Create these objects by passing SubPathInfo objects to the PathItems.add() method. This method creates a `SubPathItem` object for each SubPathInfo object, and creates and returns a new PathItem object for the path represented by all of the subpaths. Access these objects in the PathItem.subPathItems collection.

- Use the SubPathItem object to retrieve information about existing subpaths. The properties are read-only.
- Use SubPathInfo to create subpaths; the properties are writeable.

Properties

Property	Value type	What it is
`closed`	`boolean`	Read-only. True if the path is closed.
`operation`	`ShapeOperation`	Read-only. How this object behaves when it intersects another SubPathItem object. Specifies how to combine the shapes if the destination path already has a selection.
`parent`	`PathItem`	Read-only. The object's container.
`pathPoints`	`PathPoints`	Read-only. The `PathPoints` collection.
`typename`	`string`	Read-only. The class name of the referenced `SubPathItem` object.

SubPathItems

A collection of SubPathItem objects that make up a PathItem. Access this object in the PathItem.subPathItems collection property.

- Use SubPathInfo to create subpaths; the properties are writeable.
- Use the SubPathItem object to retrieve information about existing subpaths. The properties are read-only.

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the collection.
parent	PathItem	Read-only. The containing path item.
typename	string	Read-only. The class name of the referenced SubPathItems object.

TargaSaveOptions

Options for saving a document in TGA (Targa) format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
resolution	TargaBitsPerPixels	Read-write. The number of bits per pixel (default: TargaBitsPerPixels.TWENTYFOUR).
rleCompression	boolean	Read-write. True to use RLE compression (default: true).
typename	string	Read-only. The class name of the referenced TargaSaveOptions object.

TextFont

Describes a font that is available to the application. Access this object in the Application.fonts collection. For example:

```
var myFont = app.fonts.getByName("ArialMT");
```

Properties

Property	Value type	What it is
family	string	Read-only. The font family.
name	string	Read-only. The name of the font.
parent	Application	Read-only. The containing application.
postScriptName	string	Read-only. The PostScript name of the font.
style	string	Read-only. The font style.
typename	string	Read-only. The class name of the referenced TextFont object.

TextFonts

The collection of fonts available on your computer. Fonts are represented by TextFont objects. Access this object in the Application.fonts collection property. For example, this displays the number of available fonts:

```
alert(app.fonts.length);
```

Properties

Property	Value type	What it is
length	number	Read-only. The number of elements in the collection.
parent	Application	Read-only. The containing application.
typename	string	Read-only. The class name of the referenced `TextFonts` object.

Methods

Method	Parameter type	Returns	What it does
getByName (name)	string	TextFont	Gets the first element in the `TextFonts` collection with the provided name.

TextItem

The text in an ArtLayer object whose kind property is `LayerKind.TEXT`. Access this object in the ArtLayer.textItem property. For example:

```
myLayers[i].textItem.contents = "Layer in " + textArray[i] + " Set Inside "
```

Many of the properties use the UnitValue type, which combines a measurement value with a measurement unit. For information about this type, see the *JavaScript Tools Guide*.

Properties

Property	Value type	What it is
alternateLigatures	boolean	Read-write. True to use alternate ligatures. **Note:** Alternate ligatures are the same as Discretionary Ligatures. See Adobe Photoshop CS6 Help for more information.
antiAliasMethod	AntiAlias	Read-write. The method of anti aliasing to use.
autoKerning	AutoKernType	Read-write. The auto kerning option to use.
autoLeadingAmount	number [0.01..5000.00]	Read-write. The percentage to use for auto (default) leading (in points). Valid only when useAutoLeading = `true`.
baselineShift	UnitValue	Read-write. The unit value to use in the baseline offset of text.
capitalization	TextCase	Read-write. The text case.
color	SolidColor	Read-write. The text color.
contents	string	Read-write. The actual text in the layer.
desiredGlyphScaling	number [50..200]	Read-write. The desired amount by which to scale the horizontal size of the text letters. A percentage value; at 100, the width of characters is not scaled. Valid only when justification = `Justification.CENTERJUSTIFIED`, `FULLYJUSTIFIED`, `LEFTJUSTIFIED`, or `Justification.RIGHTJUSTIFIED`. When used, the minimumGlyphScaling and maximumGlyphScaling values are also required.

Property	Value type	What it is (Continued)
desiredLetterScaling	number [100..500]	Read-write. The amount of space between letters (at 0, no space is added between letters). Equivalent to **Letter Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = Justification.CENTERJUSTIFIED, FULLYJUSTIFIED, LEFTJUSTIFIED, or Justification.RIGHTJUSTIFIED. When used, the minimumLetterScaling and maximumLetterScaling values are also required.
desiredWordScaling	number [0..1000]	Read-write. The amount (percentage) of space between words (at 100, no additional space is added between words). Equivalent to **Word Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = Justification.CENTERJUSTIFIED, FULLYJUSTIFIED, LEFTJUSTIFIED, or Justification.RIGHTJUSTIFIED. When used, the minimumWordScaling and maximumWordScaling values are also required.
direction	Direction	Read-write. The text orientation.
fauxBold	boolean	Read-write. True to use faux bold (default: false). Setting this to true is equivalent to selecting text and clicking Faux Bold in the Character palette.
fauxItalic	boolean	Read-write. True to use faux italic (default: false). Setting this to true is equivalent to selecting text and clicking Faux Italic in the Character palette.
firstLineIndent	UnitValue [-1296..1296] points	Read-write. The amount (unit value) to indent the first line of paragraphs.
font	string	Read-write. The text face of the character. Use the PostScript Name of the font. See TextFont and use the postScriptName property.
hangingPunctuation	boolean	Read-write. True to use Roman hanging punctuation.
height	UnitValueX	Read-write. The height of the bounding box (unit value) for paragraph text. Valid only when kind = TextType.PARAGRAPHTEXT.
horizontalScale	number [0..1000]	Read-write. Character scaling (horizontal) in proportion to verticalScale (a percentage value).

Property	Value type	What it is (Continued)
hyphenateAfterFirst	number [1..15]	Read-write. The number of letters after which hyphenation in word wrap is allowed.
hyphenateBeforeLast	number [1..15]	Read-write. The number of letters before which hyphenation in word wrap is allowed.
hyphenateCapitalWords	boolean	Read-write. True to allow hyphenation in word wrap of capitalized words.
hyphenateWordsLongerThan	number [2..25]	Read-write. The minimum number of letters a word must have in order for hyphenation in word wrap to be allowed.
hyphenation	boolean	Read-write. True to use hyphenation in word wrap.
hyphenationZone	UnitValue [0..720] pica	Read-write. The distance at the end of a line that will cause a word to break in unjustified type.
hyphenLimit	number	Read-write. The maximum number of consecutive lines that can end with a hyphenated word.
justification	Justification	Read-write. The paragraph justification.
kind	TextType	Read-write. The text-wrap type.
language	Language	Read-write. The language to use.
leading	UnitValue	Read-write. The leading amount.
leftIndent	UnitValue [-1296..1296] points	Read-write. The amoun of space to indent text from the left.
ligatures	boolean	Read-write. True to use ligatures.
maximumGlyphScaling	number [50..200]	Read-write. The maximum amount to scale the horizontal size of the text letters (a percentage value; at 100, the width of characters is not scaled). Valid only when justification = Justification.CENTERJUSTIFIED, FULLYJUSTIFIED, LEFTJUSTIFIED, or Justification.RIGHTJUSTIFIED. When used, the minimumGlyphScaling and desiredGlyphScaling values are also required.

Property	Value type	What it is (Continued)
`maximumLetterScaling`	`number [100..500]`	Read-write. The maximum amount of space to allow between letters (at 0, no space is added between letters). Equivalent to **Letter Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = `Justification.CENTERJUSTIFIED`, `FULLYJUSTIFIED`, `LEFTJUSTIFIED`, or `Justification.RIGHTJUSTIFIED`. When used, the minimumLetterScaling and desiredLetterScaling values are also required.
`maximumWordScaling`	`number [0..1000]`	Read-write. The maximum amount of space to allow between words (a percentage value; at 100, no additional space is added between words). Equivalent to **Word Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = `Justification.CENTERJUSTIFIED`, `FULLYJUSTIFIED`, `LEFTJUSTIFIED`, or `Justification.RIGHTJUSTIFIED`. When used, the minimumWordScaling and desiredWordScaling values are also required.
`minimumGlyphScaling`	`number [50..200]`	Read-write. The minimum amount to scale the horizontal size of the text letters (a percentage value; at 100, the width of characters is not scaled). Valid only when justification = `Justification.CENTERJUSTIFIED`, `FULLYJUSTIFIED`, `LEFTJUSTIFIED`, or `Justification.RIGHTJUSTIFIED`. When used, the maximumGlyphScaling and desiredGlyphScaling values are also required.

Property	Value type	What it is (Continued)
minimumLetterScaling	number [100..500]	Read-write. The minimum amount of space to allow between letters (a percentage value; at 0, no space is removed between letters). Equivalent to **Letter Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = Justification.CENTERJUSTIFIED, FULLYJUSTIFIED, LEFTJUSTIFIED, or Justification.RIGHTJUSTIFIED. When used, the maximumLetterScaling and desiredLetterScaling values are also required.
minimumWordScaling	number [0..1000]	Read-write. The minimum amount of space to allow between words (a percentage value; at 100, no additional space is removed between words). Equivalent to **Word Spacing** in the Justification dialog (Select **Justification** on the Paragraphs palette menu). Valid only when justification = Justification.CENTERJUSTIFIED, FULLYJUSTIFIED, LEFTJUSTIFIED, or Justification.RIGHTJUSTIFIED. When used, the maximumWordScaling and desiredWordScaling values are also required.
noBreak	boolean	Read-write. True to disallow line breaks in this text. **Tip:** When true for many consecutive characters, can prevent word wrap and thus may prevent some text from appearing on the screen.
oldStyle	boolean	Read-write. True to use old style type.
parent	ArtLayer	Read-write. The containing layer.
position	array of UnitValue	Read-write. The position of origin for the text. The array members specify the X and Y coordinates. Equivalent to clicking the text tool at a point in the document to create the point of origin for text.
rightIndent	UnitValue [-1296..1296] points	Read-write. The amount of space to indent text from the right.
size	UnitValue	Read-write. The font size in UnitValue . NOTE: Type was points for CS3 and older..
spaceAfter	UnitValue [-1296..1296] points	Read-write. The amount of space to use after each paragraph.

Property	Value type	What it is (Continued)
spaceBefore	UnitValue [-1296..1296] points	Read-write. The amount of space to use before each paragraph.
strikeThru	StrikeThruType	Read-write. The text strike-through option to use.
textComposer	TextComposer	Read-write. The composition method to use to evaluate line breaks and optimize the specified hyphenation and justification options. Valid only when kind = TextType.PARAGRAPHTEXT.
tracking	number [-1000..10000]	Read-write. The amount of uniform spacing between multiple characters. Tracking units are 1/1000 of an em space. The width of an em space is relative to the current type size. In a 1-point font, 1 em equals 1 point; in a 10-point font, 1 em equals 10 points. So, for example, 100 units in a 10-point font are equivalent to 1 point.
typename	string	Read-only. The class name of the referenced textItem object.
underline	UnderlineType	Read-write. The text underlining options.
useAutoLeading	boolean	Read-write. True to use a font's built-in leading information.
verticalScale	number [0-1000]	Read-write. Vertical character scaling in proportion to horizontalScale (a percentage value).
warpBend	number [-100..100]	Read-write. The warp bend percentage.
warpDirection	Direction	Read-write. The warp direction.
warpHorizontalDistortion	number [-100..100]	Read-write. The horizontal distortion of the warp (a percentage value).
warpStyle	WarpStyle	Read-write. The style of warp to use.
warpVerticalDistortion	number [-100..100]	Read-write. The vertical distortion of the warp (a percentage value).
width	UnitValue	Read-write. The width of the bounding box for paragraph text. Valid only when kind = TextType.PARAGRAPHTEXT.

Methods

Method	Parameter type	Returns	What it does
convertToShape ()			Converts the text item and its containing layer to a fill layer with the text changed to a clipping path.
createPath ()			Creates a clipping path from the outlines of the actual text items (such as letters or words).

TiffSaveOptions

Options for saving a document in TIFF format using the Document.saveAs() method.

Properties

Property	Value type	What it is
alphaChannels	boolean	Read-write. True to save the alpha channels.
annotations	boolean	Read-write. True to save the annotations.
byteOrder	ByteOrder	Read-write. The order in which the document's multibyte values are read (default: `ByteOrder.MACOS` in Mac OS, `ByteOrder.IBM` in Windows).
embedColorProfile	boolean	Read-write. True to embed the color profile in the document.
imageCompression	TIFFEncoding	Read-write. The compression type (default: `TIFFEncoding.NONE`).
interleaveChannels	boolean	Read-write. True if the channels in the image will be interleaved.
jpegQuality	number [0..12]	Read-write. The quality of the produced image, which is inversely proportionate to the amount of JPEG compression. Valid only when imageCompression = `TIFFEncoding.JPEG`.
layerCompression	LayerCompression	Read-write. The method of compression to use when saving layers (as opposed to saving composite data). Valid only when layers = `true`.
layers	boolean	Read-write. True to save the layers.
saveImagePyramid	boolean	Read-write. True to preserve multi-resolution information (default: `false`).
spotColors	boolean	Read-write. True to save the spot colors.
transparency	boolean	Read-write. True to save the transparency as an additional alpha channel when the file is opened in another application.
typename	string	Read-only. The class name of the referenced `TiffSaveOptions` object.

UnitValue

ExtendScript defines the JavaScript class `UnitValue` to represent measurement values together with their measurement units; see 'JavaScript support in Adobe Photoshop CS6' on page 32. For references details of these classes, see the *JavaScript Tools Guide*.

xmpMetadata

Camera RAW image XMP metadata.

For camera RAW image files, the XMP metadata is stored in a *sidecar* file, which is a file in the same folder as the RAW file with the same base name and an XMP extension.

Properties

Property	Value type	What it is
parent	Document	Read-only. The containing document.
rawData	string	Read-write. A string containing the XMP metadata in XML (RDF) format. See the *XMP Specification* for details of this format.
typename	string	Read-only. The class name of the referenced xmpMetadata object.

3 JavaScript Resource

This section describes the JavaScript resource that enables your JavaScripts to behave like a plug-in. This includes:

- the ability to specify a menu the script appears in as a command,
- a terminology resource so the script can function with the Action Manager, which allows your script to record and be automated by scripting parameters,
- a category to enable ordering and grouping of commands within menus, and
- an enable string that indicates whether the command is enabled or disabled given a set of conditions.
- The strings must be valid XML syntax. The "`&`" character will not work for example. Use the `&&` for example to get logical '`&&`' in the enableinfo block.

JavaScript resource syntax

The JavaScript Resource has an HTML-style syntax, with each `<tag>` matched by a closing `</tag>`. This resource needs to appear within comments (`/* ... */`) and should be defined at the top of your script file (within the first 10,240 characters of the file.)

Tag	Description
`<javascriptresource>`	The resource definition tag.
`<name>`	The command name that appears in the Photoshop menu. If this tag is not provided in the resource, the name of the command in the menu defaults to the name of the script.
`<menu>`	The menu the command appears in. If this tag is not provided, the command appears in the **File > Scripts** menu. **Note:** Currently the only supported values for `<menu>` are `automate`, `filter` and `help`. automate puts the script in the **File > Automate** menu for example.
`<about>`	A string that appears in an About box, which the user can select from the **Help > About Plug-in** menu.
`<enableinfo>`	A boolean expression that indicates whether the command is enabled in the menu. See Enable-info grammar. **Note:** If you provide this tag, the menu item is enabled if and only if there is at least one document open, and the boolean expression evaluates to true. If you always want the menu item enabled, do not use this tag.
`<eventid>`	A unique string that identifies the event. Using a UUID will ensure that your script wont share this identifier with another script.

Tag	Description
<category>	The category the command appears within in the menu. Used to group and order commands in the menu. Commands are placed in the menu alphabetically based on the string in <category>. If two commands use the same category, they are grouped together.
<terminology>	The terminology dictionary for the script to function with the Action Manager. See the Terminology dictionary.

Basic JavaScript resource example

This example shows a very basic <javascriptresource>. With this resource, the script can be executed by selecting the command **Add a Document**, which appears in the **Automate** menu. This command is enabled in the menu, provided at least one document is already open. If the user requests information about the script from the **About Plug-in** menu, the string contained in the <about> tag is displayed in a dialog box.

```
/*
   <javascriptresource>
   <name>Add a Document</name>
   <type>automate</type>
   <about>A short string providing information about the script.</about>
   <enableinfo>true</enableinfo>
   </javascriptresource>
*/

app.documents.add();
```

Enable-info grammar

The <enableinfo> tag provides a boolean expression that, when evaluated, indicates whether the command is enabled in the menu. You can use this expression to enable or disable the menu item based on various characteristics of the document. The Enable Info grammar is as follows:

```
<booleanExpression> :=    <conjunction> { "||" <conjunction> }

<conjunction> :=          <relation> {"&&" <relation> }

<relation> :=             <equality> {<relationOperator><equality>}

<equality> :=             <simpleExpression> {<equalityOperator><simpleExpression>}

<simpleExpression> :=     <term> {<addOperator><term>}

<term> :=                 <factor> {<mulOperator><factor>}

<factor> :=               <integer> | <intrinsic> | <ident> |
                          "(" <booleanExpression ")" | "(" simpleExpression ")" |
                          "+" <factor> | "-" <factor> | "!" <factor>

<integer> :=              digit {digit}

<intrinsic> :=            <limitFunction> | <dimFunction> | <inFunction>

<limitFunction> :=        ( "min" | "max") "(" <simpleExpression> ","
                          <simpleExpression> { "," <simpleExpression> } ")"

<dimFunction> :=          "dim" "(" <simpleExpression> "," <simpleExpression> ")"
```

```
<ident> :=                  (alpha | "_") {alpha | digit | "_" }
<mulOperator> :=            "*" | "/"
<addOperator> :=            "+" | "-"
<equalityOperator> :=       "==" | "!="
<relationOperator> :=       "<" | "<=" | ">=" | ">"
<inFunction> :=             "in" "(" <simpleExpression> {"," <simpleExpression> } ")"
```

Operator precedence is shown in the following table. Operators are listed with the highest order of precedence at the top of the table

<table>
<tr><th>Operator</th><th>Description</th></tr>
<tr><td>||</td><td>Or</td></tr>
<tr><td>&&</td><td>And</td></tr>
<tr><td>+ -</td><td>Addition or subtraction</td></tr>
<tr><td>* /</td><td>Multiply or divide</td></tr>
<tr><td>< <= >= ></td><td>Less than, less than or equal, greater than or equal, greater than</td></tr>
<tr><td>== !=</td><td>Equals, or does not equal.</td></tr>
<tr><td>(..) in() max() min()
unary + - !</td><td>Functions
Unary operators: increment, decrement, not</td></tr>
</table>

The grammar provides variables and constants that you can use in the <enableinfo> expression. The following table provides a list of the constants that are available.

Constant Name	Description
true	Boolean true
false	Boolean false
BitmapMode	Bitmap mode.
GrayScaleMode	Grayscale mode, 8 bit depth.
IndexedMode	Indexed color mode.
RGBMode	RGB color mode.
CMYKMode	CMYK color mode.
HSLMode	HSL color mode.
HSBMode	HSB color mode
MultiChannelMode	Multichannel mode.
DuotoneMode	Duotone mode.
LabMode	Lab color mode.

Constant Name	Description
Gray16Mode	Grayscale mode, 16 bits per channel
RGB48Mode	RGB color mode, 16 bits per channel.
Lab48Mode	LAB mode, 16 bits per channel.
CMYK64Mode	CMYK mode, 16 bits per channel.
DeepMultichannelMode	Deep multichannel mode.
Duotone16Mode	Duotone mode, 16 bit depth.
RGB96Mode	RGB color mode, 32 bits per channel.
Gray32Mode	Grayscale mode, 32 bit depth.

The following table show the set of variables you can use in the <enableinfo> expression. The value of these variables is set based on the properties of the active document.

Variable Name	Description
PSHOP_ImageMode	Image mode of the active document.
PSHOP_ImageDepth	Depth of the active document.
PSHOP_HasLayerMask	Boolean indicating presence of layer mask.
PSHOP_HasSelectionMask	Boolean indicating presence of selection mask.
PSHOP_HasTransparencyMask	Boolean indicating presence of transparency mask.
PSHOP_NumTargetChannels	Number of target channels.
PSHOP_NumTrueChannels	Numer of image channels.
PSHOP_IsAdjustorSheet	Boolean
PSHOP_IsTargetComposite	Boolean indicating whether channels are flattened.
PSHOP_IsTargetSection	Boolean.
PSHOP_IsTargetVisible	Boolean.
PSHOP_ImageWidth	Width of the image.
PSHOP_ImageHeight	Height of the image.
PSHOP_TargetProtectFlags.	

Undefined values in enable-info evaluation

If any arithmetic or relation operation contains an operand whose value is undefined, or a variable that is undefined, the result of that evaluation is `false`.

Boolean values are treated as in C/C++, where non-zero values are `true`, and zero is `false`, with the exception that an undefined value is also `false`.

Using the "in" function

The in function (see `<inFunction>`) returns `true` is the first parameter is equal to at least one of the subsequent parameters. A typical use might be to see if the image mode of the active document is one of a set of image modes. For example:

```
in(PSHOP_ImageMode, RGBMode, CMYKMode, LabMode)
```

Action Manager automation

For your script to be able to record scripting parameters and be automated by them, it requires the addition of two basic mechanisms:

- A *terminology dictionary* that maps your script's user interface to human readable text, providing text and type information for each parameter the script uses.
- Code to read parameter information when it comes from the Action Manager, rather than from the user-interface, and code to write parameter information to the Action Manger. This code uses the Action Manager classes ActionDescriptor, ActionList, and ActionReference.

See `Conditional Mode Change.jsx` for an example of a script that can record and be automated by scripting parameters. This file can be found in the Presets/Scripts folder.

Terminology dictionary

The JavaScript resource provides a `<terminology>` tag that allows you to provide the terminology dictionary for your script. The first step in creating a terminology dictionary is to review your script's user interface, and create human-readable strings for each element in your user interface.

For example, in the Conditional Mode Change command, the user interface requests a source mode and a target mode. Both source mode and target mode have several options. All of these elements of the user interface need to have entries in the terminology dictionary.

The terminology dictionary is created in a PDF dictionary format, with the following entries, and must have the following format in the `<javascriptresource>`:

```
<terminology><![CDATA[<<<
      /Version integer
      /Events <<event dictionary>>
      /Classes <<class dictionary>>
      /Enumerations <<enumeration dictionary>>
>>> ]]></terminology>
```

Note: The information in the terminology tag needs to be wrapped in a `CDATA` block so the xml parser will ignore "/" and other tags that appear in the terminology.

The defintions for events, classes and enumerations dictionaries are provided below.

The `/Events` dictionary contains an entry for each event:

```
/eventName [                    // Name used in string-based API
   (String event name)          // required
   /direct parameter type       // optional; if omitted, no parameter
      <<                        // optional parameter dictionary
      /parameterName [          // Name used in string-based API
         (String name)          // required
         /parameter type        // required
         ]
      ...                       // other parameters
      >>
   ]
```

The `/Classes` dictionary contains and entry for each class:

```
/className [                    // Name used in string-based API
   (ZString class name)         // required
      <<                        // property dictionary
      /propertyName [           // Name used in string-based API
         (String name)          // required
         /property type         // required
         ]
      ...                       // other properties
      >>
   ]
```

The `/Enumerations` dictionary contains an entry for each enumerated type:

```
/enumTypeName                   // Name used in string-based API
   <<
   /enumValue (String name)     // required
   ...
   >>
```

Value type defintions

For `/parameter type` and `/property type` definitons, you can use the Class and Enumeration type declarations you make in your own terminology dictionary, you can use declarations provided by Photoshop or you can use basic value types.

Basic Value Types

The basic value types are shown in the following table:

Name	Code	Description
`typeInteger`	`'long'`	`int32`
`typeFloat`	`'doub'`	IEEE 64 bit double
`typeBoolean`	`'bool'`	`TRUE` or `FALSE`.
`typeText`	`'TEXT'`	Block of any number of readable characters.
`typeAlias`	`'alis'`	Macintosh file system path.

Name	Code	Description
typePaths	'Pth '	Windows file system path.
typePlatformFilePath	'alis' or'Pth	typeAlias for Mac OS, typePath for Windows.

Predefined Class Types

Photoshop provides a number of predefined classes that are available for use in the terminology dictionary. A useful subset of those classes is shown in the table below. Use these classes when they are appropriate, but you can define new classes in the terminology resource, if necessary.

Name	Code	Description
classColor	'Clr '	Class for color classes.
classRGBColor	'RGBC'	keyRed, keyGreen, keyBlue
classCMYKColor	'CMYC'	keyCyan, keyMagenta, keyYellow, keyBlack.
classUnspecifiedColor	'UnsC'	Unspecified.
classGrayscale	'Grsc'	keyGray
classBookColor	'BkCl '	Book color
classLabColor	'LbCl'	keyLuminance, keyA, keyB.

Uniqueness rules for terminology entries

Generally, the names for terminology entries must be unique within a particular category and scope. It is best to not make names unique unnecessarily; generic terms are prefereable, and if a name already exists for something, go ahead and use it. Case matters in considering uniqueness of terminology entries.

The uniqueness rules for terminology entries are:

- All event names must be different from all other event names.
- All class names must be different from all other names.
- All enumeration type names must be different from all other enumeration type names.
- All keys must be different from all other keys used in the same class or event.
- All enumeration values must be different from all other enumeration values in the same enumeration type.
- A class, event, enumeration type, key, and enumeration value can all have the same name.

Terminology definition example

This example demonstrates the terminology definition for a new event; the example uses ZStrings. The event is called `newAnnot`, and it takes three parameters:

- `annotType`, an enumeration (`annotType`)
- `at`, a class (`point`), and
- `size`, a class (`annotSizeClass`).

The `annotSizeClass` has two properties: `width`, and `height`, both of type `floatType`. The enumeration `annotType` has three values: `annotUnknown`, `annotText`, and `annotSound`.

```
<terminology><![CDATA[<<<
   /Version 1
   /Events
      /newAnnot [(New Annotation) <<
         /annotType [(Type) /annotType]
         /at  [(At) /Point]
         /size  [(Size) /annotSizeClass] >>]
   /Classes
      /annotSizeClass [(Size) <<
         /width [(Width) /floatType]
         /height [(Height) /floatType]
         >>]
   /Enumerations
      /annotType <<
         /annotUnknown (Unknown)
         /annotText (Text)
         /annotSound (Sound)
   >>> ]]></terminology>
```

4 Scripting Constants

This section lists and describes the enumerations defined for use with Adobe Photoshop CS6 JavaScript properties and methods.

Constant type	Values	What it means
AdjustmentReference	ABSOLUTE RELATIVE	Method to use for interpreting selective color adjustment specifications: ABSOLUTE = % of the whole. RELATIVE = % of the existing color amount. Pass to ArtLayer.selectiveColor().
AnchorPosition	BOTTOMCENTER BOTTOMLEFT BOTTOMRIGHT MIDDLECENTER MIDDLELEFT MIDDLERIGHT TOPCENTER TOPLEFT TOPRIGHT	The point around which to transform an object. This is the point that does not move when an object is rotated or resized using methods in ArtLayer, LayerSet, and Selection, or when the entire canvas is resized with Document.resizeCanvas().
AntiAlias	CRISP NONE SHARP SMOOTH STRONG	Method to use to smooth edges by softening the color transition between edge pixels and background pixels. Used in a TextItem.antiAliasMethod.
AutoKernType	MANUAL METRICS OPTICAL	The type of kerning to use for characters. Used in TextItem.autoKerning.
BatchDestinationType	FOLDER NODESTINATION SAVEANDCLOSE	The destination, if any, for batch-processed files, specified in the BatchOptions used with the Application.batch() method: FOLDER: Save modified versions of the files to a new location (leaving the originals unchanged). NODESTINATIONTYPE: Leave all files open. SAVEANDCLOSE: Save changes and close the files.

Constant type	Values	What it means
BitmapConversionType	CUSTOMPATTERN DIFFUSIONDITHER HALFTHRESHOLD HALFTONESCREEN PATTERNDITHER	Specifies the quality of an image you are converting to bitmap mode. Used in BitmapConversionOptions.
BitmapHalfToneType	CROSS DIAMOND ELLIPSE LINE ROUND SQUARE	Specifies the shape of the dots (ink deposits) in the halftone screen. Used in BitmapConversionOptions.
BitsPerChannelType	EIGHT ONE SIXTEEN THIRTYTWO	The number of bits per color channel. Value of Document.bitsPerChannel; pass to Documents.add(). Also used in PDFOpenOptions and CameraRAWOpenOptions.
BlendMode	COLORBLEND COLORBURN COLORDODGE DARKEN DIFFERENCE DISSOLVE DIVIDE EXCLUSION HARDLIGHT HARDMIX HUE LIGHTEN LINEARBURN LINEARDODGE LINEARLIGHT LUMINOSITY MULTIPLY NORMAL OVERLAY PASSTHROUGH PINLIGHT SATURATION SCREEN SOFTLIGHT SUBTRACT VIVIDLIGHT	Controls how pixels in an image are blended when a filter is applied. The value of ArtLayer.blendMode and LayerSet.blendMode.
BMPDepthType	BMP_A1R5G5B5 BMP_A4R4G4B4 BMP_A8R8G8B8 BMP_R5G6B5 BMP_R8G8B8 BMP_X1R5G5B5 BMP_X4R4G4B4 BMP_X8R8G8B8 EIGHT FOUR ONE SIXTEEN THIRTYTWO TWENTYFOUR	The number of bits per channel (also called pixel depth or color depth). The number selected indicates the exponent of 2. For example, a pixel with a bit-depth of EIGHT has 2^8, or 256, possible color values. Used in BMPSaveOptions.

Constant type	Values	What it means
ByteOrder	IBM MACOS	The platform-specific order in which multibyte values are read.
CameraRAWSettingsType	CAMERA CUSTOM SELECTEDIMAGE	The default CameraRaw settings to use: the camera settings, custom settings, or the settings of the selected image. Set in CameraRAWOpenOptions.
CameraRAWSize	EXTRALARGE LARGE MAXIMUM MEDIUM MINIMUM SMALL	The camera RAW size type options:. EXTRALARGE=5120 x 4096 LARGE=4096 x 2731 MAXIMUM=6144 X 4096 MEDIUM=3072 x 2048 MINIMUM=1536 x 1024 SMALL=2048 x 1365 Set in CameraRAWOpenOptions.
ChangeMode	BITMAP CMYK GRAYSCALE INDEXEDCOLOR LAB MULTICHANNEL RGB	The new color profile or mode for a document, specified in Document.changeMode(). **Note:** Color images must be changed to GRAYSCALE mode before you can change them to BITMAP mode.
ChannelType	COMPONENT MASKEDAREA SELECTEDAREA SPOTCOLOR	The type of a color channel: COMPONENT: related to document color mode. MASKEDAREA: Alpha channel where color indicates masked area. SELECTEDAREA: Alpha channel where color indicates selected are. SPOTCOLOR: Alpha channel to store a spot color.

Constant type	Values	What it means
ColorBlendMode	BEHIND CLEAR COLOR COLORBURN COLORDODGE DARKEN DARKERCOLOR DIFFERENCE DISSOLVE EXCLUSION HARDLIGHT HARDMIXBLEND HUE LIGHTEN LIGHTERCOLOR LINEARBURN LINEARDODGE LINEARLIGHT LUMINOSITY MULTIPLY NORMAL OVERLAY PINLIGHT SATURATION SCREEN SOFTLIGHT VIVIDLIGHT	The way color should be blended in a fill or stroke operation. Pass to PathItem.fillPath(), Selection.fill(), Selection.stroke()
ColorModel	CMYK GRAYSCALE HSB LAB NONE RGB	The color model to use for a SolidColor.
ColorPicker	ADOBE APPLE PLUGIN WINDOWS	The preferred color-selection tool, set in Preferences.
ColorProfileType	CUSTOM NONE WORKING	The type of color profile used to manage this document, set in Document.colorProfileType.
ColorReductionType	ADAPTIVE BLACKWHITE CUSTOM GRAYSCALE MACINTOSH PERCEPTUAL RESTRICTIVE SELECTIVE WINDOWS	The color reduction algorithm option for ExportOptionsSaveForWeb.
ColorSpaceType	ADOBERGB COLORMATCHRGB PROPHOTORGB SRGB	The type of color space to use in CameraRAWOpenOptions.
CopyrightedType	COPYRIGHTEDWORK PUBLICDOMAIN UNMARKED	The copyright status of a document. Used in DocumentPrintSettings.copyrighted.

Constant type	Values	What it means
`CreateFields`	DUPLICATION INTERPOLATION	The method to use for creating fields. Pass to ArtLayer.applyDeInterlace().
`CropToType`	ARTBOX BLEEDBOX BOUNDINGBOX CROPBOX MEDIABOX TRIMBOX	The style to use when cropping a page in a PDF document. Set in PDFOpenOptions.cropPage.
`DCSType`	COLORCOMPOSITE GRAYSCALECOMPOSITE NOCOMPOSITE	The type of composite DCS file to create with DCS1_SaveOptions or DCS2_SaveOptions: COLORCOMPOSITE: Creates a color composite file in addition to DCS files. GRAYSCALECOMPOSITE: Creates a grayscale composite file in addition to DCS files. NOCOMPOSITE: Does not create a composite file.
`DepthMapSource`	IMAGEHIGHLIGHT LAYERMASK NONE TRANSPARENCYCHANNEL	The source to use for the depth map. Pass to ArtLayer.applyLensBlur().
`DescValueType`	ALIASTYPE BOOLEANTYPE CLASSTYPE DOUBLETYPE ENUMERATEDTYPE INTEGERTYPE LARGEINTEGERTYPE LISTTYPE OBJECTTYPE RAWTYPE REFERENCETYPE STRINGTYPE UNITDOUBLE	The value type of an action key, returned by ActionDescriptor.getType() and ActionList.getType().
`DialogModes`	ALL ERROR NO	Controls the type of dialogs Photoshop displays when running scripts.
`Direction`	HORIZONTAL VERTICAL	• The direction in which to flip the document canvas, passed to Document.flipCanvas(). • The orientation of text in TextItem.direction. • The direction of text warping in TextItem.warpDirection.

Constant type	Values	What it means
DisplacementMapType	STRETCHTOFIT TILE	Describes how the displacement map fits the image if the image is not the same size as the map. Pass to ArtLayer.applyDisplace().
Dither	DIFFUSION NOISE NONE PATTERN	The type of dithering to use in GIFSaveOptions, IndexedConversionOptions and ExportOptionsSaveForWeb.
DocPositionStyle	PRINTCENTERED USERDEFINED	The type of positioning to use in DocPosition
DocumentFill	BACKGROUNDCOLOR TRANSPARENT WHITE	The fill type of a new document, passed to Documents.add().
DocumentMode	BITMAP CMYK DUOTONE GRAYSCALE INDEXEDCOLOR LAB MULTICHANNEL RGB	The color mode of a open document, Document.mode. See also Document.changeMode().
EditLogItemsType	CONCISE DETAILED SESSIONONLY	The preferred level of detail in th history log, set in Preferences: CONCISE: Save a concise history log. DETAILED: Save a detailed history log. SESSIONONLY: Save history log only for the session.
ElementPlacement	INSIDE PLACEATBEGINNING PLACEATEND PLACEBEFORE PLACEAFTER	The object's position in the Layers palette. **Note:** Not all values are valid for all object types. See the specific object description to make sure you are using a valid value.
EliminateFields	EVENFIELDS ODDFIELDS	The type of fields to eliminate. Pass to ArtLayer.applyDeInterlace().
ExportType	ILLUSTRATORPATHS SAVEFORWEB	The type of export for Document.exportDocument(). This is equivalent to choosing **File > Export > Paths To Illustrator**, or **File > Save For Web and Devices**.
Extension	LOWERCASE NONE UPPERCASE	The policy and format for appending an extension to the filename when saving with Document.saveAs().

Constant type	Values	What it means
FileNamingType	DDMM DDMMYY DOCUMENTNAMELOWER DOCUMENTNAMEMIXED DOCUMENTNAMEUPPER EXTENSIONLOWER EXTENSIONUPPER MMDD MMDDYY SERIALLETTERLOWER SERIALLETTERUPPER SERIALNUMBER1 SERIALNUMBER2 SERIALNUMBER3 SERIALNUMBER4 YYDDMM YYMMDD YYYYMMDD	File naming options for the BatchOptions used with the Application.batch() method.
FontPreviewType	HUGE EXTRALARGE LARGE MEDIUM NONE SMALL	The preferred type size to use for font previews in the type tool font menus , set in Preferences.
FontSize	LARGE MEDIUM SMALL	The preferred type size to use for panels and dialogs, set in Preferences.
ForcedColors	BLACKWHITE NONE PRIMARIES WEB	The type of colors to be included the color table regardless of their usage. Used in GIFSaveOptions and IndexedConversionOptions. BLACKWHITE: Pure black and pure white. NONE: None PRIMARIES: Red, green, blue, cyan, magenta, yellow, black, and white. WEB: the 216 web-safe colors.
FormatOptions	OPTIMIZEDBASELINE PROGRESSIVE STANDARDBASELINE	The option with which to save a JPEG file, in JPEGSaveOptions. OPTIMIZEDBASELINE: Optimized color and a slightly reduced file size. PROGRESSIVE: Displays a series of increasingly detailed scans as the image downloads. STANDARDBASELINE: Format recognized by most web browsers.
GalleryConstrainType	CONSTRAINBOTH CONSTRAINHEIGHT CONSTRAINWIDTH	The type of proportions to constrain for images. Used in GalleryImagesOptions.

Constant type	Values	What it means
GalleryFontType	ARIAL COURIERNEW HELVETICA TIMESNEWROMAN	The fonts to use for the Web photo gallery captions and other text. Used in GalleryBannerOptions, GalleryImagesOptions, and GalleryThumbnailOptions. Also used in PicturePackageOptions.
GallerySecurityTextColorType	BLACK CUSTOM WHITE	The color to use for text displayed over gallery images as an antitheft deterrent. Used in GallerySecurityOptions.
GallerySecurityTextPositionType	CENTERED LOWERLEFT LOWERRIGHT UPPERLEFT UPPERRIGHT	The position of the text displayed over gallery images as an antitheft deterrent. Used in GallerySecurityOptions. Also used in PicturePackageOptions.
GallerySecurityTextRotateType	CLOCKWISE45 CLOCKWISE90 COUNTERCLOCKWISE45 COUNTERCLOCKWISE90 ZERO	The orientation of the text displayed over gallery images as an antitheft deterrent. Used in GallerySecurityOptions. Also used in PicturePackageOptions.
GallerySecurityType	CAPTION COPYRIGHT CREDIT CUSTOMTEXT FILENAME NONE TITLE	The content to use for text displayed over gallery images as an antitheft deterrent. Used in GallerySecurityOptions. **Note:** All types draw from the image's file information except CUSTOMTEXT.
GalleryThumbSizeType	CUSTOM LARGE MEDIUM SMALL	The size of thumbnail images in the web photo gallery. Used in GalleryThumbnailOptions.
Geometry	HEPTAGON HEXAGON OCTAGON PENTAGON SQUARE TRIANGLE	Geometric options for shapes, such as the iris shape in the Lens Blur Filter. Pass to ArtLayer.applyLensBlur().
GridLineStyle	DASHED DOTTED SOLID	The preferred line style for the nonprinting grid displayed over images, set in Preferences.
GridSize	LARGE MEDIUM NONE SMALL	The preferred size of grid line spacing, set in Preferences.

Constant type	Values	What it means
GuideLineStyle	DASHED SOLID	The preferred line style for nonprinting guides displayed over images, set in Preferences.
IllustratorPathType	ALLPATHS DOCUMENTBOUNDS NAMEDPATH	The paths to export to an Illustrator file using Document.exportDocument().
Intent	ABSOLUTECOLORIMETRIC PERCEPTUAL RELATIVECOLORIMETRIC SATURATION	The rendering intent to use when converting from one color space to another with Document.convertProfile() or Document.print()
Justification	CENTER CENTERJUSTIFIED FULLYJUSTIFIED LEFT LEFTJUSTIFIED RIGHT RIGHTJUSTIFIED	The placement of paragraph text within the bounding box. Used in TextItem.justification.
Language	BRAZILLIANPORTUGUESE CANADIANFRENCH DANISH DUTCH ENGLISHUK ENGLISHUSA FINNISH FRENCH GERMAN ITALIAN NORWEGIAN NYNORSKNORWEGIAN OLDGERMAN PORTUGUESE SPANISH SWEDISH SWISSGERMAN	The language to use for text. Used in TextItem.language.
LayerCompression	RLE ZIP	Compression methods for data for pixels in layers, when saving to TIFF format. Used in TiffSaveOptions.

Constant type	Values	What it means
LayerKind	BLACKANDWHITE BRIGHTNESSCONTRAST CHANNELMIXER COLORBALANCE CURVES EXPOSURE GRADIENTFILL GRADIENTMAP HUESATURATION INVERSION LEVELS NORMAL PATTERNFILL PHOTOFILTER POSTERIZE SELECTIVECOLOR SMARTOBJECT SOLIDFILL TEXT THRESHOLD LAYER3D VIBRANCE VIDEO	The type of a layer object, in ArtLayer.kind. **Note:** You can create a text layer only from an empty art layer.
LensType	MOVIEPRIME PRIME105 PRIME35 ZOOMLENS	The type of lens to use. Pass to ArtLayer.applyLensFlare().
MagnificationType	ACTUALSIZE FITPAGE	The type of magnification to use when viewing an image. Used in PresentationOptions.
MatteType	BACKGROUND BLACK FOREGROUND NETSCAPE NONE SEMIGRAY WHITE	The color to use to fill anti-aliased edges adjacent to transparent areas of the image. When transparency is turned off for an image, the matte color is applied to transparent areas. Used in GIFSaveOptions, IndexedConversionOptions, and JPEGSaveOptions.
MeasurementRange	ALLMEASUREMENTS ACTIVEMEASUREMENTS	The measurement to act upon. Pass to MeasurementLog methods.
MeasurementSource	MEASURESELECTION MEASURECOUNTTOOL MEASURERULERTOOL	The source for recording measurements. Pass to Document.recordMeasurements().
NewDocumentMode	BITMAP CMYK GRAYSCALE LAB RGB	The color profile to use for a new document. Pass to Documents.add(). Also used in ContactSheetOptions and PicturePackageOptions.

Constant type	Values	What it means
NoiseDistribution	GAUSSIAN UNIFORM	Distribution method to use when applying an Add Noise filter. Pass to ArtLayer.applyAddNoise().
OffsetUndefinedAreas	REPEATEDGEPIXELS SETTOBACKGROUND WRAPAROUND	Method to use to fill the empty space left by offsetting a an image or selection. Pass to ArtLayer.applyOffset().
OpenDocumentMode	CMYK GRAYSCALE LAB RGB	The color profile to use when opening an EPS or PDF document. Pass to `app.open()` in EPSOpenOptions or PDFOpenOptions.
OpenDocumentType	ALIASPIX BMP CAMERARAW COMPUSERVEGIF DICOM ELECTRICIMAGE EPS EPSPICTPREVIEW EPSTIFFPREVIEW FILMSTRIP JPEG PCX PDF PHOTOCD PHOTOSHOP PHOTOSHOPDCS_1 PHOTOSHOPDCS_2 PHOTOSHOPEPS PHOTOSHOPPDF PICTFILEFORMAT PICTRESOURCEFORMAT PIXAR PNG PORTABLEBITMAP RAW SCITEXCT SGIRGB SOFTIMAGE TARGA TIFF WAVEFRONTRLA WIRELESSBITMAP	The format in which to open the document, using `app.open()`. **Note:** `PHOTOCD` is deprecated. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS6 Install DVD. **Note:** The `DICOM` option is for the Extended version only.
OperatingSystem	OS2 WINDOWS	The target operating system in BMPSaveOptions.
Orientation	LANDSCAPE PORTRAIT	Page orientation for PhotoCDOpenOptions, deprecated in Photoshop CS3. **Note:** Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS6 Install DVD.

Constant type	Values	What it means
OtherPaintingCursors	PRECISEOTHER STANDARDOTHER	The preferred pointer for the following tools: Eraser, Pencil, Paintbrush, Healing Brush, Rubber Stamp, Pattern Stamp, Smudge, Blur, Sharpen, Dodge, Burn, Sponge. Set in Preferences.
PaintingCursors	BRUSHSIZE PRECISE STANDARD	The preferred pointer for the following tools: Marquee, Lasso, Polygonal Lasso, Magic Wand, Crop, Slice, Patch Eyedropper, Pen, Gradient, Line, Paint Bucket, Magnetic Lasso, Magnetic Pen, Freeform Pen, Measure, Color Sampler. Set in Preferences.
PaletteType	EXACT LOCALADAPTIVE LOCALPERCEPTUAL LOCALSELECTIVE MACOSPALETTE MASTERADAPTIVE MASTERPERCEPTUAL MASTERSELECTIVE PREVIOUSPALETTE UNIFORM WEBPALETTE WINDOWSPALETTE	The palette type to use in GIFSaveOptions and IndexedConversionOptions.
PathKind	CLIPPINGPATH NORMALPATH TEXTMASK VECTORMASK WORKPATH	The type of a PathItem.
PDFCompatibility	PDF13 PDF14 PDF15 PDF16 PDF17	The PDF version to make the document compatible with. Used in PDFSaveOptions.
PDFEncoding	JPEG JPEG2000HIGH JPEG2000LOSSLESS JPEG2000LOW JPEG2000MED JPEG2000MEDHIGH JPEG2000MEDLOW JPEGHIGH JPEGLOW JPEGMED JPEGMEDHIGH JPEGMEDLOW NONE PDFZIP PDFZIP4BIT	The type of compression to use when saving a document in PDF format. Used in PDFSaveOptions.

Constant type	Values	What it means
PDFResample	NONE PDFAVERAGE PDFBICUBIC PDFSUBSAMPLE	The down sample method to use. Used in PDFSaveOptions.
PDFStandard	NONE PDFX1A2001 PDFX1A2003 PDFX32002 PDFX32003 PDFX42008	The PDF standard to make the document compatible with. Used in PDFSaveOptions.
PhotoCDColorSpace	LAB16 LAB8 RGB16 RGB8	The color space for PhotoCDOpenOptions, deprecated in Photoshop CS3. **Note:** Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS6 Install DVD.
PhotoCDSize	EXTRALARGE LARGE MAXIMUM MEDIUM MINIMUM SMALL	The pixel dimensions of the image in PhotoCDOpenOptions, deprecated in Photoshop CS3. EXTRALARGE = 1024x1536 LARGE = 512x768 MAXIMUM = 2048x3072 MEDIUM = 256x384 MINIMUM = 64x96 SMALL = 128x192 **Note:** Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS6 Install DVD.
PICTBitsPerPixels	EIGHT FOUR SIXTEEN THIRTYTWO TWO	The number of bits per pixel to use when compression a PICT file. Used in PICTFileSaveOptions and PICTResourceSaveOptions. **Note:** Use 16 or 32 for RGB images; use 2, 4, or 8 for bitmap and grayscale images.
PICTCompression	JPEGHIGHPICT JPEGLOWPICT JPEGMAXIMUMPICT JPEGMEDIUMPICT NONE	The type of compression to use when saving an image as a PICT file. Used in PICTFileSaveOptions and PICTResourceSaveOptions.
PicturePackageTextType	CAPTION COPYRIGHT CREDIT FILENAME NONE ORIGIN USER	The function or meaning of text in a Picture Package. Used in PicturePackageOptions.

Constant type	Values	What it means
PointKind	CORNERPOINT SMOOTHPOINT	The role a PathPoint plays in a PathItem.
PointType	POSTSCRIPT TRADITIONAL	The preferred measurement to use for type points, set in Preferences.pointSize: POSTSCRIPT = 72 points/inch. TRADITIONAL = 72.27 points/inch.
PolarConversionType	POLARTORECTANGULAR RECTANGULARTOPOLAR	The method of polar distortion to use. Pass to ArtLayer.applyPolarCoordinates().
Preview	EIGHTBITTIFF MACOSEIGHTBIT MACOSJPEG MACOSMONOCHROME MONOCHROMETIFF NONE	The type of image to use as a low-resolution preview in the destination application. Used in DCS1_SaveOptions, DCS2_SaveOptions, and EPSSaveOptions.
PrintColorHandling	PRINTERMANAGED PHOTOSHOPMANAGED SEPARATIONS	The type of color handling to use for ColorHandling
PurgeTarget	ALLCACHES CLIPBOARDCACHE HISTORYCACHES UNDOCACHES	Cache to be targeted in an Application.purge() operation.
QueryStateType	ALWAYS ASK NEVER	The preferred policy for checking whether to maximize compatibility when opening PSD files, set in Preferences.maximizeCompatibility.
RadialBlurMethod	SPIN ZOOM	The blur method to use. Pass to ArtLayer.applyRadialBlur().
RadialBlurQuality	BEST DRAFT GOOD	The smoothness or graininess of the blurred image. Pass to ArtLayer.applyRadialBlur().
RasterizeType	ENTIRELAYER FILLCONTENT LAYERCLIPPINGPATH LINKEDLAYERS SHAPE TEXTCONTENTS	The layer element to rasterize, using ArtLayer.rasterize().
ReferenceFormType	CLASSTYPE ENUMERATED IDENTIFIER INDEX NAME OFFSET PROPERTY	The type of an ActionReference object, returned by getForm().

Constant type	Values	What it means
ResampleMethod	BICUBIC BICUBICSHARPER BICUBICSMOOTHER BILINEAR NEARESTNEIGHBOR NONE	The method to use for image interpolation. Passed to Document.resizeImage(), and used as the value of Preferences.interpolation.
RippleSize	LARGE MEDIUM SMALL	The size of undulations to use. Pass to ArtLayer.applyRipple().
SaveBehavior	ALWAYSSAVE ASKWHENSAVING NEVERSAVE	The application's preferred behavior when saving a document. See Preferences.appendExtension and imagePreviews
SaveDocumentType	ALIASPIX BMP COMPUSERVEGIF ELECTRICIMAGE JPEG PCX PHOTOSHOP PHOTOSHOPDCS_1 PHOTOSHOPDCS_2 PHOTOSHOPEPS PHOTOSHOPPDF PICTFileFORMAT PICTRESOURCEFORMAT PIXAR PNG PORTABLEBITMAP RAW SCITEXCT SGIRGB SOFTIMAGE TARGA TIFF WAVEFRONTRLA WIRELESSBITMAP	The format in which to save a document when exporting with Document.exportDocument(). Pass in ExportOptionsSaveForWeb.format, to to specify the type of file to write. Only the following are supported for export: COMPUSERVEGIF, JPEG, PNG-8, PNG-24, and BMP.
SaveEncoding	ASCII BINARY JPEGHIGH JPEGLOW JPEGMAXIMUM JPEGMEDIUM	The type of encoding to use when saving a file to DCS or EPS with Document.saveAs().
SaveLogItemsType	LOGFILE LOGFILEANDMETADATA METADATA	The preferred location of history log data, set in Preferences.saveLogItems.
SaveOptions	DONOTSAVECHANGES PROMPTTOSAVECHANGES SAVECHANGES	The policy for closing a document with Document.close().

Constant type	Values	What it means
`SelectionType`	DIMINISH EXTEND INTERSECT REPLACE	The selection behavior when a selection already exists: DIMINISH: Remove the selection from the already selected area. EXTEND: Add the selection to an already selected area. INTERSECT: Make the selection only the area where the new selection intersects the already selected area. REPLACE: Replace the selected area. Used in PathItem.makeSelection(), Selection.load(), Selection.select(), and Selection.store().
`ShapeOperation`	SHAPEADD SHAPEINTERSECT SHAPESUBTRACT SHAPEXOR	How to combine the shapes if the destination path already has a selection. Set for SubPathInfo.operation, stored in the resulting SubPathItem.
`SmartBlurMode`	EDGEONLY NORMAL OVERLAYEDGE	The method to use for smart blurring: EDGEONLY, OVERLAYEDGES: Apply blur only to edges of color transitions. NORMAL: Apply blur to entire image. Pass to ArtLayer.applySmartBlur().
`SmartBlurQuality`	HIGH LOW MEDIUM	The blur quality to use. Pass to ArtLayer.applySmartBlur().
`SourceSpaceType`	DOCUMENT PROOF	The color space for source when printing with Document.print().
`SpherizeMode`	HORIZONTAL NORMAL VERTICAL	The curve (or stretch shape) to use for the distortion. Pass to ArtLayer.applySpherize().
`StrikeThruType`	STRIKEBOX STRIKEHEIGHT STRIKEOFF	The style of strikethrough to use in text. Used in TextItem.strikeThru.
`StrokeLocation`	CENTER INSIDE OUTSIDE	The placement of path or selection boundary strokes. Pass to Selection.stroke().
`TargaBitsPerPixels`	SIXTEEN THIRTYTWO TWENTYFOUR	The resolution to use when saving an image in Targa format. Used in TargaSaveOptions.

Constant type	Values	What it means
TextCase	ALLCAPS NORMAL SMALLCAPS	The capitalization style to use in text. Used in TextItem.capitalization.
TextComposer	ADOBEEVERYLINE ADOBESINGLELINE	The composition method to use to optimize the specified hyphenation and justification options. Used in TextItem.textComposer.
TextType	PARAGRAPHTEXT POINTTEXT	The type of text, used in TextItem.kind. PARAGRAPHTEXT: Text that wraps within a bounding box. POINTTEXT: Text that does not wrap.
TextureType	BLOCKS CANVAS FILE FROSTED TINYLENS	The type of texture or glass surface image to load for a texturizer or glass filter. Pass to ArtLayer.applyGlassEffect().
TIFFEncoding	JPEG NONE TIFFLZW TIFFZIP	The type of compression to use for TIFF files. Used in TiffSaveOptions.
ToolType	ARTHISTORYBRUSH BACKGROUNDERASER BLUR BRUSH BURN CLONESTAMP COLORREPLACEMENTTOOL DODGE ERASER HEALINGBRUSH HISTORYBRUSH PATTERNSTAMP PENCIL SHARPEN SMUDGE SPONGE	The tool to use with PathItem.strokePath().
TransitionType	BLINDSHORIZONTAL BLINDSVERTICAL BOXIN BOXOUT DISSOLVE GLITTERDOWN GLITTERRIGHT GLITTERRIGHTDOWN NONE RANDOM SPLITHORIZONTALIN SPLITHORIZONTALOUT SPLITVERTICALIN SPLITVERTICALOUT WIPEDOWN WIPELEFT WIPERIGHT WIPEUP	The method to use for transition from one image to the next in a PDF presentation. Used in PresentationOptions.

Constant type	Values	What it means
TrimType	BOTTOMRIGHT TOPLEFT TRANSPARENT	Type of pixels to trim around an image, passed to Document.trim().: BOTTOMRIGHT = bottom right pixel color. TOPLEFT = top left pixel color.
TypeUnits	MM PIXELS POINTS	The preferred unit for text character measurements, set in Preferences.
UndefinedAreas	REPEATEDGEPIXELS WRAPAROUND	The method to use to treat undistorted areas or areas left blank in an image to which the a filter in the Distort category has been applied. Pass to ArtLayer.applyDisplace(), applyShear(), applyWave().
UnderlineType	UNDERLINELEFT UNDERLINEOFF UNDERLINERIGHT	The placement of text underlining. Used in TextItem.underline. **Note:** UNDERLINELEFT and UNDELINERIGHT are valid only when direction = Direction.VERTICAL.
Units	CM INCHES MM PERCENT PICAS PIXELS POINTS	The preferred measurement unit for type and ruler increments, set in Preferences.rulerUnits.
Urgency	FOUR HIGH LOW NONE NORMAL SEVEN SIX THREE TWO	The editorial urgency status of a document, set in DocumentPrintSettings.urgency.
WarpStyle	ARC ARCH ARCLOWER ARCUPPER BULGE FISH FISHEYE FLAG INFLATE NONE RISE SHELLLOWER SHELLUPPER SQUEEZE TWIST WAVE	The warp style to use for text. Used in TextItem.warpStyle.

Constant type	Values	What it means
WaveType	SINE SQUARE TRIANGULAR	The type of wave to use. Pass to ArtLayer.applyWave().
WhiteBalanceType	ASSHOT AUTO CLOUDY CUSTOM DAYLIGHT FLASH FLUORESCENT SHADE TUNGSTEN	Lighting conditions that affect color balance. Set in CameraRAWOpenOptions.
ZigZagType	AROUNDCENTER OUTFROMCENTER PONDRIPPLES	The method of zigzagging to use. Pass to ArtLayer.applyZigZag().

Appendix A: Event ID Codes

The following table lists events and their four-character ID codes or string identifiers for use with the `notifier` object.

Note: **Do not include single quotes (') with four-character IDs in your code**. The single quotes are used in this table to illustrate the placement of required spaces in codes that do not contain four letters. However, string identifiers, which are longer than four characters, require double quotes in the code.

Tip: If you can't find the event you want to use for notification in this table, you can use ScriptListener to determine the event ID code. See the ScriptListener documentation in the Action Manager chapter of the *Photoshop CS6 Scripting Guide*.

Event	4-char ID or String
`3DTransform`	`'TdT '`
`Average`	`'Avrg'`
`ApplyStyle`	`'ASty'`
`Assert`	`'Asrt'`
`AccentedEdges`	`'AccE'`
`Add`	`'Add '`
`AddNoise`	`'AdNs'`
`AddTo`	`'AddT'`
`Align`	`'Algn'`
`All`	`'All '`
`AngledStrokes`	`'AngS'`
`ApplyImage`	`'AppI'`
`BasRelief`	`'BsRl'`
`Batch`	`'Btch'`
`BatchFromDroplet`	`'BtcF'`
`Blur`	`'Blr '`
`BlurMore`	`'BlrM'`
`Border`	`'Brdr'`
`Brightness`	`'BrgC'`
`CanvasSize`	`'CnvS'`
`ChalkCharcoal`	`'ChlC'`
`ChannelMixer`	`'ChnM'`

Event	4-char ID or String
Charcoal	'Chrc'
Chrome	'Chrm'
Clear	'Cler'
Close	'Cls '
Clouds	'Clds'
ColorBalance	'ClrB'
ColorHalftone	'ClrH'
ColorRange	'ClrR'
ColoredPencil	'ClrP'
ContactSheet	"0B71D221-F8CE-11d2-B21B-0008C75B322C"
ConteCrayon	'CntC'
Contract	'Cntc'
ConvertMode	'CnvM'
Copy	'copy'
CopyEffects	'CpFX'
CopyMerged	'CpyM'
CopyToLayer	'CpTL'
Craquelure	'Crql'
CreateDroplet	'CrtD'
Crop	'Crop'
Crosshatch	'Crsh'
Crystallize	'Crst'
Curves	'Crvs'
Custom	'Cstm'
Cut	'cut '
CutToLayer	'CtTL'
Cutout	'Ct '
DarkStrokes	'DrkS'
DeInterlace	'Dntr'
DefinePattern	'DfnP'
Defringe	'Dfrg'
Delete	'Dlt '
Desaturate	'Dstt'

Event	4-char ID or String
Deselect	'Dslc'
Despeckle	'Dspc'
DifferenceClouds	'DrfC'
Diffuse	'Dfs '
DiffuseGlow	'DfsG'
DisableLayerFX	'dlfx'
Displace	'Dspl'
Distribute	'Dstr'
Draw	'Draw'
DryBrush	'DryB'
Duplicate	'Dplc'
DustAndScratches	'DstS'
Emboss	'Embs'
Equalize	'Eqlz'
Exchange	'Exch'
Expand	'Expn'
Export	'Expr'
Jumpto	'Jpto'
ExportTransparentImage	"02879e00-cb66-11d1-bc43-0060b0a13dc4"
Extrude	'Extr'
Facet	'Fct '
Fade	'Fade'
Feather	'Fthr'
Fibers	'Fbrs'
Fill	'Fl '
FilmGrain	'FlmG'
Filter	'Fltr'
FindEdges	'FndE'
FitImage	"3caa3434-cb67-11d1-bc43-0060b0a13dc4"
FlattenImage	'FltI'
Flip	'Flip'
Fragment	'Frgm'
Fresco	'Frsc'

Event	4-char ID or String
GaussianBlur	'GsnB'
Get	'getd'
Glass	'Gls '
GlowingEdges	'GlwE'
Gradient	'Grdn'
GradientMap	'GrMp'
Grain	'Grn '
GraphicPen	'GraP'
Group	'GrpL'
Grow	'Grow'
HalftoneScreen	'HlfS'
Hide	'Hd '
HighPass	'HghP'
HSBHSL	'HsbP'
HueSaturation	'HStr'
ImageSize	'ImgS'
Import	'Impr'
InkOutlines	'InkO'
Intersect	'Intr'
IntersectWith	'IntW'
Inverse	'Invs'
Invert	'Invr'
LensFlare	'LnsF'
Levels	'Lvls'
LightingEffects	'LghE'
Link	'Lnk '
Make	'Mk '
Maximum	'Mxm '
Median	'Mdn '
MergeLayers	'Mrg2'
MergeLayersOld	'MrgL'
MergeSpotChannel	'MSpt'
MergeVisible	'MrgV'

Event	4-char ID or String
Mezzotint	'Mztn'
Minimum	'Mnm '
ModeChange	"8cba8cd6-cb66-11d1-bc43-0060b0a13dc4"
Mosaic	'Msc '
Mosaic_PLUGIN	'MscT'
MotionBlur	'MtnB'
Move	'move'
NTSCColors	'NTSC'
NeonGlow	'NGlw'
Next	'Nxt '
NotePaper	'NtPr'
Notify	'Ntfy'
Null	typeNull
OceanRipple	'OcnR'
Offset	'Ofst'
Open	'Opn '
Paint	'Pnt '
PaintDaubs	'PntD'
PaletteKnife	'PltK'
Paste	'past'
PasteEffects	'PaFX'
PasteInto	'PstI'
PasteOutside	'PstO'
Patchwork	'Ptch'
Photocopy	'Phtc'
PicturePackage	"4C1ABF40-DD82-11d2-B20F-0008C75B322C"
Pinch	'Pnch'
Place	'Plc '
Plaster	'Plst'
PlasticWrap	'PlsW'
Play	'Ply '
Pointillize	'Pntl'
Polar	'Plr '

Event	4-char ID or String
PosterEdges	'PstE'
Posterize	'Pstr'
Previous	'Prvs'
Print	'Prnt'
ProfileToProfile	'PrfT'
Purge	'Prge'
Quit	'quit'
RadialBlur	'RdlB'
Rasterize	'Rstr'
RasterizeTypeSheet	'RstT'
RemoveBlackMatte	'RmvB'
RemoveLayerMask	'RmvL'
RemoveWhiteMatte	'RmvW'
Rename	'Rnm '
ReplaceColor	'RplC'
Reset	'Rset'
ResizeImage	"1333cf0c-cb67-11d1-bc43-0060b0a13dc4"
Reticulation	'Rtcl'
Revert	'Rvrt'
Ripple	'Rple'
Rotate	'Rtte'
RoughPastels	'RghP'
Save	'save'
Select	'slct'
SelectiveColor	'SlcC'
Set	'setd'
SharpenEdges	'ShrE'
Sharpen	'Shrp'
SharpenMore	'ShrM'
Shear	'Shr '
Show	'Shw '
Similar	'Smlr'
SmartBlur	'SmrB'

Event	4-char ID or String
Smooth	'Smth'
SmudgeStick	'SmdS'
Solarize	'Slrz'
Spatter	'Spt '
Spherize	'Sphr'
SplitChannels	'SplC'
Sponge	'Spng'
SprayedStrokes	'SprS'
StainedGlass	'StnG'
Stamp	'Stmp'
Stop	'Stop'
Stroke	'Strk'
Subtract	'Sbtr'
SubtractFrom	'SbtF'
Sumie	'Smie'
TakeMergedSnapshot	'TkMr'
TakeSnapshot	'TkSn'
TextureFill	'TxtF'
Texturizer	'Txtz'
Threshold	'Thrs'
Tiles	'Tls '
TornEdges	'TrnE'
TraceContour	'TrcC'
Transform	'Trnf'
Trap	'Trap'
Twirl	'Twrl'
Underpainting	'Undr'
Undo	'undo'
Ungroup	'Ungr'
Unlink	'Unlk'
UnsharpMask	'UnsM'
Variations	'Vrtn'
Wait	'Wait'

Event	4-char ID or String
WaterPaper	'WtrP'
Watercolor	'Wtrc'
Wave	'Wave'
Wind	'Wnd '
ZigZag	'ZgZg'
BackLight	'BacL'
FillFlash	'FilE'
ColorCast	'ColE'

Index

D

Q

R

S

T

U